



Computer Science
Faculty of Natural Sciences and Engineering
Åbo Akademi University
Åbo 2020

MASTERS THESIS

On the Applicability of Network Science in Personalized Medicine

Supervisor at UTU:

Prof. ION PETRE

Supervisor at ÅAU:

Dr. EUGEN CZEIZLER

Author: ELIO NUSHI (1800222)

Acknowledgments

I would like to express my gratitude to my supervisor from the University of Turku Ion Petre for introducing me to the field of computational biology. His help and guidance were crucial for the successful completion of this thesis. Ion, you have been a mentor to me through the whole process of my Master's studies and you will always remain a role model who I will follow for the rest of my life.

I wish to show my appreciation to my supervisor from Åbo Akademi Eugen Czeizler. Thank you Eugen for the invaluable explanation on the theory of graphs and string algorithms, your dedication and pedagogical style of teaching full of mnemonics makes me remember almost every lecture that we had.

I am deeply grateful to Åbo Akademi which gave me the unique opportunity to study on a full scholarship and assigned me the excellence award to fund my stay and studies for the whole continuation of my Master's degree. Without it this whole experience would not have been possible. I have tried my best to justify the trust and responsibility that I was given.

Finally, I would like to thank my family for the emotional support that they have given me in the endeavours to succeed in my studies and career away from home. Special thanks also go to my brother Ani who persistently asks me to join him in the US and start up a business in Silicon Valley, his motivational speeches have given me determination to follow my dreams.

Thank you all again,
Turku/Åbo, April 2020

Abstract

The availability of big data regarding genetic information and the knowledge about the behavior and interactions between genes and proteins have drawn the interest of computational scientists in the field of biology. In particular system biology is the science which aims to study the complex communication between objects in biological environments in order to get a holistic understanding of living systems as opposed to the reductionist approach which studies the components separately. Mathematical models are usually built in order to analyze these complex biological systems such as protein-protein interactions (PPI) and disease networks. The ultimate goal of understanding the system is being able to manipulate it into a desired state which is referred to as the controllability of the system. System controllability is a strong background motivation for the concept of personalized medicine or precision medicine which aims to identify treatment lines based on individual characteristics of the patients in order to find the most appropriate drug(s) which can transition the biological system from a sick state to a healthy state by minimizing side effects.

Full controllability over a network can be solved in polynomial time but the solutions that it offers especially in the case of complex systems is large, thus making it inappropriate to use for personalized medicine. In the case of cancer networks which are considerably large and complex having full controllability is not useful given the big number of nodes that have to be changed by external controllers (e.g, drugs). Since full controllability offers infeasible solutions to use in practice, a more realistic goal is to obtain target controllability - that is, being able to transform the network from an initial state to a new state where only some of the nodes have the desired values (i.e., target nodes). Target controllability has been proven to be a *NP-Complete* problem and many approximate computational techniques have been tried to solve it.

In this thesis we focus on the core intuition behind some of the approximate techniques of solving target controllability whose aim is to keep the set of drug target nodes as small as possible. We exploit several network science methods based on centrality measures to approach the problem of gaining information over biological networks in terms of their topological structure and the identification of important nodes. Numerous studies have been conducted to analyze the concept of centrality in the context of social networks. However, their possible applicability on biological networks has not taken equal attention. We thought it is relevant and necessary to provide a comprehensive summary of the commonly used centrality methods and see how well they predict important proteins and genes in

cancer networks. Furthermore, we review the topic of random graphs, discuss their properties, and describe different models that exist for generating them. Afterwards, we identify common properties that real multiple myeloma (MM) cancer networks share with random graph models. Finally, we apply different centrality methods in MM networks and compare the outcomes with what is already known from clinical medicine and supported by research papers in the field. Our final goal is to identify the significant genes and proteins that play a crucial role in the development of this disease based merely on topological attributes.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Related Works	4
2	Mathematical Preliminaries	6
2.1	Algorithm Analysis and Complexity	6
2.2	Networks and Graph Theory	9
3	Centrality Methods as Ranking Algorithms	13
3.1	Degree Centrality	16
3.2	Closeness Centrality	20
3.3	Betweenness Centrality	25
3.4	Harmonic Centrality	30
3.5	Degree Prestige	32
3.6	Eigenvector-based Prestige	34
3.7	Katz Prestige	39
4	Random Graphs	43
4.1	Statistical Properties	44
4.2	Small-World	48
4.2.1	Watts-Strogatz Model	51
4.3	Scale-Free	52
4.3.1	Barabási-Albert Model	55
4.3.2	Barabási-Ravasz-Vicsek Model	56
4.4	Random Graphs Generation	58
5	Methodologies Used	61
5.1	Description and Analysis of Data	61
5.2	Centrality Analysis of the Networks	63

5.2.1	MM-0028 Network	64
5.2.2	MM-0038 Network	65
5.2.3	MM-0191 Network	65
5.3	Interpretations and Conclusion	66
6	Discussion	68
	Bibliography	70
	Appendix A Centrality Analysis Results for MM Networks	76
A.1	MM-0028 Network	76
A.2	MM-0038 Network	79
A.3	MM-0191 Network	81

Chapter 1

Introduction

Networks are excellent tools for modeling different phenomena spanning from telecommunication, traffic flow control, biological structures, to social analysis and many others. The first record of such usage is found in the works of the mathematician Leonhard Euler who used an unprecedented abstraction to model and solve the problem of the bridges of Königsberg [1], thus setting the foundations of the branch of mathematics called graph theory. Since then it has been observed that networks can be used to simplify and help with the analysis of many problems coming from various fields of science such as biology, chemistry, physics, engineering, and sociology. Network science is the study of network representations of physical, biological, and social phenomena, leading to predictive models of these phenomena [2]. Network science emerged as an interdisciplinary field which uses concepts and methods from graph theory, statistics, algorithms and applies them to build models that can be used to study and understand complex systems.

1.1 Motivation

The ultimate technologies for sequencing the DNA have generated high amounts of data and revealed important information regarding the genetic structure of living cells. Advances in experimental data acquisition have contributed to the discovery of properties and functions of genes, RNA, proteins and their interactions [3]. These interactions allow for the signaling pathways and other biological processes which make the cell perform its normal functions [3]. The communication between components of the nucleus within a cell and external influences (e.g., drugs, proteins) can be modeled by using directed graphs consisting of nodes which

represent proteins and directed edges describing the type of interaction between the nodes. Weights can be assigned to each edge in order to show the strength of the interaction between proteins. It makes sense to associate variables to each of the nodes which represent the level of activation of the proteins depending on the incoming edges and the level of activation of the predecessor nodes. Any node is influenced by incoming edges and predecessors in the same way as it can also influence successor nodes via its outgoing edges. At the end, we obtain a dynamical system where given an initial set of values to the nodes of the network, a change in some of the nodes will be propagated and cause changes to the rest of the nodes in the network. Depending on the set in which we allow variables to take values from we distinguish between continuous dynamical systems which are generally solved with ordinary differential equations (ODE) and discrete dynamical systems which model of computation can be built using difference equations or Boolean functions.

The analysis of interaction between cellular components using dynamical systems has been one of the main focus of biological research in the recent years, hence allowing new understandings to the field of molecular biology from the point of view of networks [3]. This networking approach has provided a framework to study and analyze the behavior of diseases such as Alzheimer, diabetes, and cancers [3].

Biology is a field in which extensive use of graph theoretical and network concepts are made. Networks offer great modeling tools for biological assemblies. It is natural to represent cell components as the nodes of a graph that are connected by edges which indicate their interoperability. These modelings occur in many studies of biology. Figure 1.1 shows the network of functional modules of organelles of a cell under drought conditions. Functional modules are groups of genes which are involved in biological transactions of signals such as protein-protein interactions (PPI) or functional associations [4]. Many biological processes like Krebs cycle can be visualized using networks. In Figure 1.2 we can see how Krebs cycle can be modeled using the different components (e.g., enzymes, metabolites) involved in it as graph's nodes and the relations among them as edges [5]. Networks are also used in neuroscience by mapping data collected via EEG or MRI to measure the activity and connection of different areas of the brain into graphs. Analyzing those graphs might yield important information on the development of neurological diseases, and their early identification as suggested by Vecchio et al. [6].

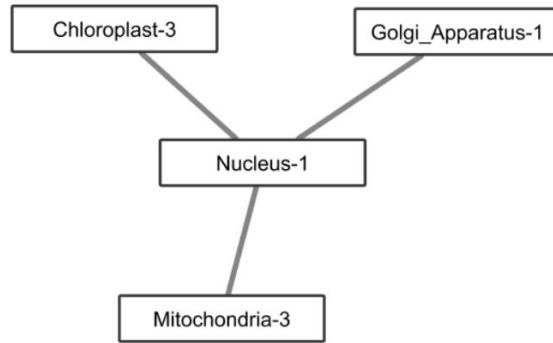


Figure 1.1: Connection between modules of organelles in a cell. Taken from [4].

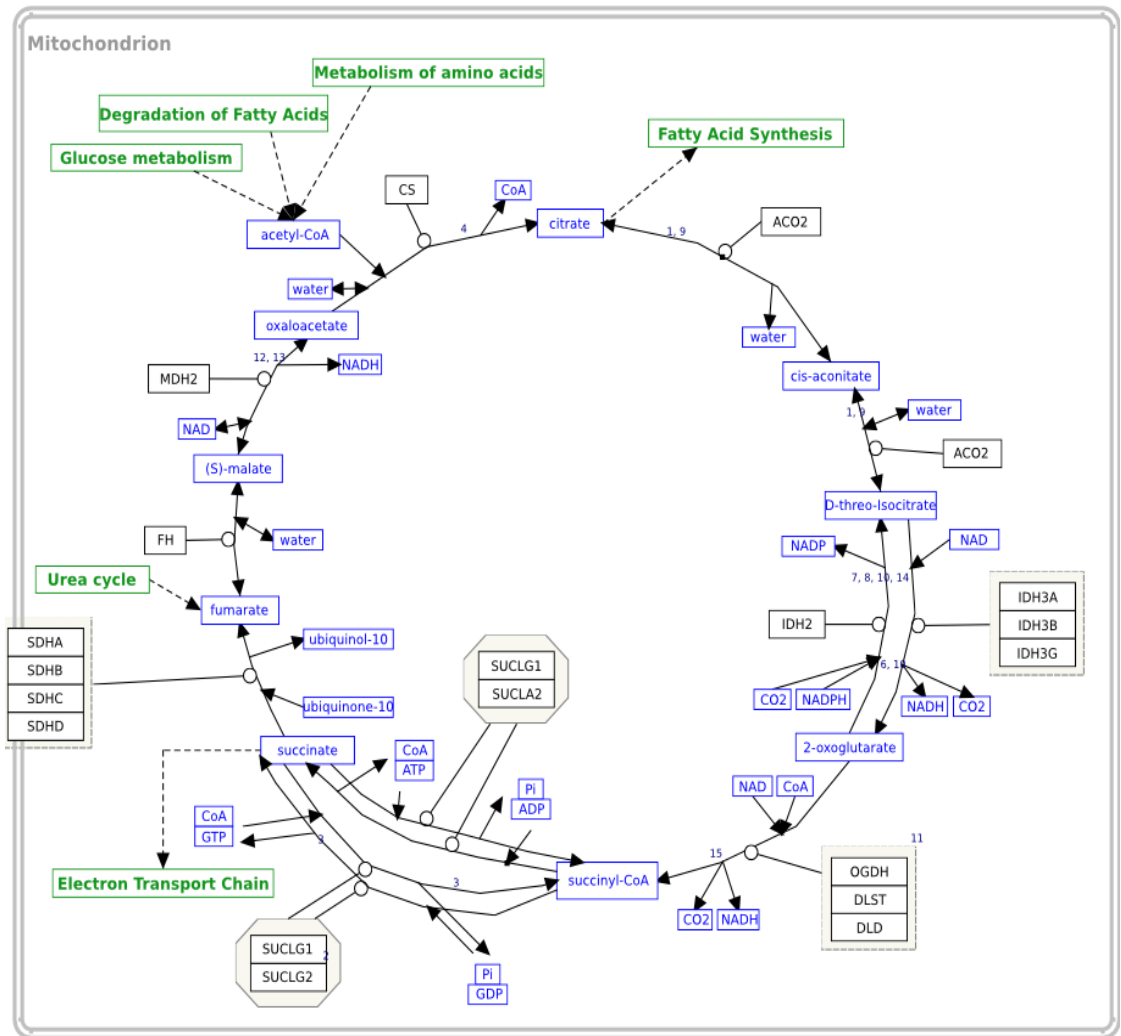


Figure 1.2: TCA cycle (i.e., Krebs or citric acid cycle). Taken from [7].

In this thesis, we take an approximate approach in trying to identify important nodes in the network which might play a significant role in its controllability by using various node ranking methods. The application of these methods in the case of multiple myeloma (MM) cancer networks is done and the determination of the genes corresponding to the highest scoring nodes is performed. Additionally, we try to discover common patterns and properties that MM cancer networks share with random graph models. Conclusively, we evaluate our results by comparing the findings with those obtained by previous methods and discuss the efficacy of our procedure.

1.2 Related Works

Centrality measures offer a way to introduce order relations in the sets of graph components. These methods have drawn the interest of many researchers who have tried to find correlations between centrality scores of nodes and other features that the objects represented by nodes manifest in the system being modeled by the network. The application of centrality methods in social sciences is well known but multiple research works on their applicability in analyzing biological systems also exist. In this section we explain some of the previously mentioned researchers and discuss them chronologically starting from the early usage in social networks to later applications in biology.

The idea of applying centrality methods for the sake of understanding importance and functionality of nodes was first proposed by Bavelas in the context of social networks [8]. The works of Bavelas, Barret and Leavit in applying centrality measures in social networks suggested that they uncover information regarding structural organization of groups and efficiency in problem solving [8]. These measures were applied in different cases, for example Cohen et al. used them to study the political integration in Indian social life, Pitts studied centralities as indication of significance for communication paths in civic growth, Beauchamp suggested that organizations can be made more efficient if their sub-units directly communicate with each other at their most central nodes, and Rogers concluded that the centrality of an organization is a function of its internal characteristics and the network to which it belongs [8].

Innovative applications of centralities were made and existing methods were further elaborated as a result of the development of biology and genetics. Hahn et al. [9] examined PPI networks in 3 eukaryote organisms and concluded that pro-

teins having higher central positions with respect to degree centrality and closeness centrality are more essential for survival and evolve slowly. In another study by Ozgur et al. [10] degree, eigenvector, betweenness and closeness centrality were successfully used to identify gene-disease association on a literature mined gene-interaction network. On a study for the identification of important proteins and regulatory pathways in PPI networks related to essential hypertension Ran et al. [11] used degree, betweenness and closeness centrality to identify and suggest the implication of the protein NOS3 in blood pressure variations. Joy et al. [12] discovered the abundant presence of proteins with high betweenness centrality while maintaining their degree centrality low in a yeast proteome network and discuss the evolutionary and functional significance of this finding.

Chapter 2

Mathematical Preliminaries

In this chapter we shortly give a formal mathematical explanation for the concepts that are frequently used in the rest of the thesis. The first section is a succinct and partial overview of algorithms analysis in terms of time and space complexity measures, as well as computational complexity focused on the notions of complexity classes. In the second section we will gently immerse in the field of graph theory and go through some basic definitions of terms and properties related to them.

2.1 Algorithm Analysis and Complexity

Algorithms are detailed step-by-step methods for solving problems; for a particular problem there generally exist multiple algorithms that can solve it. Depending on the implementation of the algorithms and the type of the problem that they are aiming to solve, comparisons have to be made in order to find the best algorithms to use for solving that problem; this is the scope of algorithm analysis. Computers are primarily our tools for executing algorithms and it is reasonable to let them ‘decide’ which algorithms are better than others in terms of two basic computer resources that they consume: memory space and processor time. However this approach is difficult to standardize since it highly depends on the computer that we are running the algorithms on, and different computers use non-identical amount of resources for the execution of the same algorithm based on their CPU power and caching strategy for instance. In order to obtain a unified and generalized way of comparing algorithms independently from the machine they are running on and its intrinsic architecture, the number of basic computer steps that they perform

with respect to the size of the input is counted as a measure of evaluation [13]. The former measurement is simplified even further by introducing the concepts of *big-O*, *big-Ω* and *big-Θ* of functions symbolically written as $O(f(n))$, $\Omega(f(n))$, $\Theta(f(n))$ respectively, where $f : \mathbb{N} \mapsto \mathbb{R}$.

Definition 2.1.1 (Big- O). *Let $f : \mathbb{N} \mapsto \mathbb{R}$ and $g : \mathbb{N} \mapsto \mathbb{R}$ be two functions. Then $f = O(g)$ if and only if there exist the constants $c > 0$ and n_0 such that $f(n) \leq c \cdot g(n) \forall n \geq n_0$.*

Writing $f(n) = O(g(n))$ is a weak analog to stating that $f(n) \leq g(n)$, because it does not necessarily satisfy the ‘ \leq ’ condition for some initial values of n or without being multiplied by a constant c as Definition 2.1.1 specifies. For example $100n = O(n^2)$ is true even though for $n = 1$ obviously it doesn’t satisfy the ‘ \leq ’ condition, but for values of $n \geq n_0 = 100$ we can safely write that $100n \leq n^2$. Seemingly we can argue that $100n = O(n^2)$ because there exists a constant, namely $c = 100$ which makes the statement $100n \leq c \cdot n^2$ hold for $n \geq n_0 = 1$. Another interpretation for $f(n) = O(g(n))$ is the following: Given a fixed n_0 there exist a constant c such that $c \cdot g(n)$ is an upper-bound function for $f(n)$. A symmetric notion exists also for lower-bound functions and is defined as follows:

Definition 2.1.2 (Big- Ω). *Let $f : \mathbb{N} \mapsto \mathbb{R}$ and $g : \mathbb{N} \mapsto \mathbb{R}$ be two functions. Then $f = \Omega(g)$ if and only if there exist the constants $c > 0$ and n_0 such that $f(n) \geq c \cdot g(n) \forall n \geq n_0$.*

Considering the definitions for upper-bound and lower-bound we can easily combine them to obtain the exact bound of a function:

Definition 2.1.3 (Big- Θ). *Let $f : \mathbb{N} \mapsto \mathbb{R}$ and $g : \mathbb{N} \mapsto \mathbb{R}$ be two functions. Then $f = \Theta(g)$ if and only if $f = O(g)$ and $f = \Omega(g)$.*

The big- O notation allows for major simplification when dealing with functions involving many terms such as $5n^3 + 2n^2 + 10n + 5$, in this case we can just ignore all minor terms and say that $5n^3 + 2n^2 + 10n + 5 = O(n^3)$ because the cubic term of the polynomial dominates all the others. The following rules are used to simplify functions when determining their big- O :

1. Constant coefficients can be omitted: for example $1000n^3$ can be written as just n^3 .
2. n^x dominates n^y if $x > y$.

3. In general exponential functions dominate over all polynomial functions, and all exponential functions with lower basis.
4. Polynomial functions dominate over logarithmic functions. For example n dominates over $(\log n)^5$, accordingly n^3 dominates over $n^2 \log n$.

The omission of multiplicative constants when determining the big- O does not mean that they are not important when designing algorithms. In fact improving the performance of an algorithm by a factor of two is considered a very plausible result. However, big- O allows for the understanding of algorithms in a more general level and has major implications from the theoretical point of view.

It is crucial to find an efficient algorithm when solving a particular problem - that is, an algorithm whose big- O is as small as possible. The extent of efficiency that an algorithm can have is bounded by the intrinsic complexity of the problem that it is trying to solve. Computational complexity is an area of research in the field of theoretical computer science which aims to classify computational problems into classes depending on their level of difficulty. Many computational classes have been defined and it is beyond the scope of this thesis to introduce them all. We consider it relevant to mention 3 very important classes, namely P , NP , and NP -complete because the starting motivation for our work is the belonging of the target controllability problem to the later class. The definition of complexity classes is typically done in the context of *decision problems* or *search problems*. We will define here the complexity classes in terms of *search problems* as given in the material presented by [13].

Definition 2.1.4 (Decision Problem). *A decision problem is an algorithmic question that can be answered by yes or no.*

Definition 2.1.5 (Search Problem). *A search problem is specified by a Boolean returning algorithm C that takes two inputs, an instance I and a proposed solution S , and runs in polynomial time in I . We say S is a solution to I if and only if $C(I, S) = \text{true}$.*

Definition 2.1.6 (P Class). *The class of all search problems that can be solved in polynomial time is denoted by P .*

Definition 2.1.7 (NP Class). *The class of all search problems is denoted by NP .*

Definition 2.1.8 (Search Problem Reduction). *A reduction from search problem A to search problem B is a polynomial-time algorithm f that transforms any instance I of A into an instance $f(I)$ of B , together with another polynomial-time algorithm h that maps any solution S of $f(I)$ back into a solution $h(S)$ of I .*

Definition 2.1.9 (NP-Complete Class). *A search problem belongs to NP-complete class if every other problem in NP can be reduced to one of its instances.*

It follows from the definitions above that $P \subseteq NP$. An interesting question is whether the statement $NP \subseteq P$ is also true, which would imply that $P = NP$. This is also known as the P vs NP open problem. Proving that $P = NP$ will have major implications in science since it would mean that exponential complexity can always be avoided, which most of algorithm researchers believe that it is highly unlikely.

2.2 Networks and Graph Theory

A *network* is a very general concept which can be defined as a set of connected objects. These objects are usually referred to as *nodes* or *vertices*, and can be visualized as points in the plane. We call *edges* the connections between these objects in the network and represent them as lines between points. In mathematics these structures are called *graphs* and are the topic of study in a separate field of mathematics which is graph theory. In this thesis we will use the terms network and graph as synonyms to refer to the same object.

Definition 2.2.1 (Graph). *Let V be a set of vertices (also called nodes) and $E \subseteq V \times V$ a set of edges between the nodes. We define a directed graph G to be the pair (V, E) and denote it as $G(V, E)$. If E is a symmetric set (i.e., $(u, v) \in E$ whenever $(v, u) \in E$) then G is called an undirected graph. In this case we indicate the edge between u and v by $\{u, v\}$ instead of (u, v) and (v, u) .*

Figure 2.1 represents a graph $G(V, E)$, where $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ and $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$. The number of vertices and edges of a graph are denoted by $|V|$ and $|E|$ accordingly. Edges can also be specified as pairs of vertices that they connect. If an edge e connects vertices u and v it can be written as $e = (u, v)$. Thus the graph in Figure 2.1 being undirected is alternatively written as $G(V, E)$ where $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\}$ and $E = \{\{v_1, v_4\}, \{v_1, v_2\}, \{v_4, v_2\}, \{v_4, v_3\}, \{v_3, v_8\}, \{v_8, v_5\}, \{v_5, v_6\}, \{v_6, v_7\}, \{v_7, v_8\}\}$. If

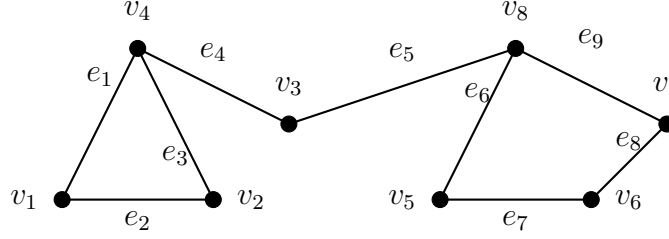


Figure 2.1: An undirected graph $G(V, E)$ with 8 vertices and 9 edges.

there exists an edge e between vertices v and u then we say that e is *incident* with u and v , also that u and v are *adjacent* (i.e., *neighbor*) to each other. If two edges have a vertex in common then we call those edges *adjacent*. In Figure 2.1 edges e_5 , e_6 , and e_9 are incident to v_8 and incident to each other while v_8 is incident to v_3 , v_5 and v_7 . If $e = (u, v)$ is an edge of a directed graph then u is called its *initial vertex* while v its *terminal vertex*. Directed edges create a *predecessor-successor* (i.e., *ancestor-descendant*) *relationships* between vertices - that is, if $e = (u, v)$ then u is called the *predecessor* (or *ancestor*) of v with respect to e while v is the *successor* (or *descendant*) of u with respect to e . Notice that we draw the edges of the graph in Figure 2.1 as line segments because it is undirected otherwise they are usually drawn as arrows pointing from the predecessor to the successor. The visual representation of graphs by drawing points and lines to connect them in a 2-dimensional space is not ideal for computational purposes. Consequently very often graphs are written in the form of *adjacency matrices*.

Definition 2.2.2 (Adjacency Matrix). *Let $G(V, E)$ be a graph, its adjacency matrix A is defined to be a $|V| \times |V|$ matrix, such that:*

$$A(i, j) = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{if } (v_i, v_j) \notin E. \end{cases}$$

The adjacency matrix of an undirected graph is symmetric, whereas in the case of a directed graph it is generally not. Considering the definition above the corresponding adjacency matrix of the graph in Figure 2.1 will be:

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}.$$

We can associate a real number w_e (i.e., *weight* of the edge) to each edge e of a graph and we call the graph together with its weights a *weighted graph*. An *unweighted graph* is considered to be a special case of a weighted graph where all weights are equal to one. Weighted graphs are useful in many application where we use graphs for modeling objects. For example weights in a graph representing communication links might show the cost of maintaining those links. Whenever we consider a graph without specifying whether it is weighted or directed, an unweighted and undirected graph is usually assumed. Below are provided the definitions of important terms regarding graphs which we make use of in this thesis:

Definition 2.2.3 (Degree). *Let $G(V, E)$ be a graph, the degree of a vertex $v \in V$ denoted by $\deg(v)$ is the number of edges incident with it.*

In Figure 2.1 $\deg(v_1) = \deg(v_2) = \deg(v_3) = \deg(v_5) = \deg(v_6) = \deg(v_7) = 2$. It can be the case for a vertex v to have $\deg(v) = 0$ and we call that vertex *isolated*. In directed graph we distinguish between *in-degree* and *out-degree* of a vertex.

Definition 2.2.4 (In-Degree). *Let $G(V, E)$ be a directed graph, the in-degree of a vertex $v \in V$ denoted by $\deg^-(v)$ is the number of ordered pairs $(v_i, v_j) \in E$ where $v_i, v_j \in V$ and $v_j = v$.*

Definition 2.2.5 (Out-Degree). *Let $G(V, E)$ be a directed graph, the out-degree of a vertex $v \in V$ denoted by $\deg^+(v)$ is the number of ordered pairs $(v_i, v_j) \in E$ where $v_i, v_j \in V$ and $v_i = v$.*

Definition 2.2.6 (Path). *Let $G(V, E)$ be a graph and $v_i, v_j \in V$, the path from vertices v_i to v_j is a sequence $P = (v_i = v_0), v_1, v_2, \dots, v_\ell, (v_{\ell+1} = v_j)$ of different*

vertices in V (except possibly for the first and the last) such that there exists $e = (v_k, v_{k+1}) \in E$ for every $0 \leq k < \ell + 1$.

Definition 2.2.7 (Length of Path). *The length of the path $P = (v_i = v_0), v_1, v_2, \dots, v_\ell, (v_{\ell+1} = v_j)$ is the sum of the weights associated to the edges determined by the pairs (v_k, v_{k+1}) for every $0 \leq k < \ell + 1$.*

A path between the vertices of a graph does not always exist and even when it does there might be several of them. If a path exists between two vertices, then those vertices are called *connected*. The *shortest path* between two vertices is a path which has the shortest length out of all the paths which connect them.

Definition 2.2.8 (Distance of Vertices). *Let $G(V, E)$ be a graph and $v_i, v_j \in V$, the distance between v_i and v_j is denoted by $d(v_i, v_j)$ and defined as the length of the shortest path that connects those nodes. If no such path exists then $d(v_i, v_j) = \infty$.*

Definition 2.2.9 (Strongly Connected Graph). *Let $G(V, E)$ be a graph, if no pair of vertices $v_i, v_j \in V$ exists such that $d(v_i, v_j) = \infty$, then the graph is called *strongly connected*.*

Definition 2.2.10 (Diameter of a Graph). *The longest shortest path between all pairs of vertices of a graph is defined to be the diameter of the graph.*

Definition 2.2.11 (Clique). *Let $G(V, E)$ be a graph, a subset $W \subseteq V$ is called a clique with size $|W|$ if any two of its elements are adjacent to each other.*

The problem of finding a clique of a given size in a graph is *NP-Complete*.

Chapter 3

Centrality Methods as Ranking Algorithms

The concept of *network centrality* has been studied for many years and has found applications in different areas. There is a common agreement among researchers that it can reveal important information regarding the structure of a network and that it is related to different attributes that the network has. However, the interpretation that is given to the results that centrality yields is very different depending on the context to which it is applied.

The concept of network centrality is very general and does not have a unique definition. Many methods have been developed to make it concrete and computable in a network. One way of thinking about the centrality of a node is by visualizing the center of a star graph. The object positioned in the center of this network is inherently perceived as the most important. In other words, the centrality measures try to answer the following question: How important is a particular node for the preservation of the overall structural properties (e.g., average degree, average distance of nodes, connectivity) in a given network? Depending on the specific method that is applied to measure the centrality of nodes, several interpretations can be made, for example nodes with higher scores are more independent, they have more control over the network, and their level of activity is higher.

While exploring different centrality methods we will indeed notice that the center node of the star (see Figure 3.2) appears to be special (i.e., unique) in many ways and they all treat it exceptionally which is a good reason for calling them centrality methods. Some of the methods are very similar to each other, yet

each of them is original in their mathematical formulations and the reasons which motivated their creation. Others are completely different and their application is intended only in specific areas. Not all the methods are intuitive to understand, nor it is clear the correct interpretation of the measurement inside the system which is modeled by the network. All the methods that we will consider are based on structural properties of the graphs such as nodes' relations and weights of edges. As we will see, these methods are calculated by taking a graph as input which is represented in the form of an adjacency matrix for symbolical simplicity. We will first explain each of the methods by putting the focus on individual nodes and then indicate how they can be applied to calculate *network centrality indices* so that they can give information about the network structure as a whole by allowing for comparisons among different networks. Depending on the similarities that are shared between them centrality methods can be grouped in 4 general classes: *Degree Centralities* (e.g., *degree centrality*), *Path Centralities* (e.g., *betweenness centrality*), *Proximity Centralities* (e.g., *closeness centrality*, *harmonic centrality*), and *Spectral Centralities* (e.g., *eigenvector centrality*, *Katz centrality*). Another categorization made is based on whether edge direction is taken or not into consideration, thus we distinguish between two terms: *centrality score* and *prestige score*. Centrality score evaluates the node considering its overall connections and disregards their direction. On the other hand, prestige score considers whether the edges attached to a node are incoming or outgoing. Incoming edges are often interpreted as measure of support while outgoing edges are considered as potential of influence or command.

In this chapter a comprehensive review of the most representative methods inside each class is made and appropriate examples are given for the sake of illustrating their applicability. Furthermore we will try to shed light on each method by describing: the motivation behind them, the formulas to compute them, the complexity of their algorithms, several alternative approximate algorithms which can be used for faster computations, their contextual applications, and the properties of the system which they expose. Particularly we will focus on a running example based on the Padget's Florentine families graph representing the marriage relations among the wealthiest families of Florence in the time of Renaissance around 1430 (see Figure 3.1). The dataset is publicly accessible and it is also available as an R package [14]. Since the measures of prestige are meant to be applied specifically in the case of directed graphs, we transformed the Padget's Florentine families graph by adding an extra edge for every existing edge

and gave them opposite directions. Thus, the graph is treated as directed when calculating the measures of prestige. In the case when measures of centrality are calculated, the graph is treated as undirected where each edge is counted twice. To enhance the readability of Padget’s Florentine families graph we have omitted the extra added edge in all visualizations of the graph throughout this thesis. We have used Python and *Networkx* library to read and enrich, with centrality and prestige scores, the graphml file representing the marital relations in the Padget’s Florentine families that we generated with R. Networkx [49] is an open source Python library used for the exploration and analysis of networks and their algorithms. All the visualizations are done in *Cytoscape* [15] that is an open source software for analyzing and visualizing complex networks.

Edges can also be subjects of centrality measurements, but that is not done very frequently. Most of the time the focus is put on the nodes because the concept can be easily translated among the two. The methods that we will consider here apply to the nodes instead of the edges.

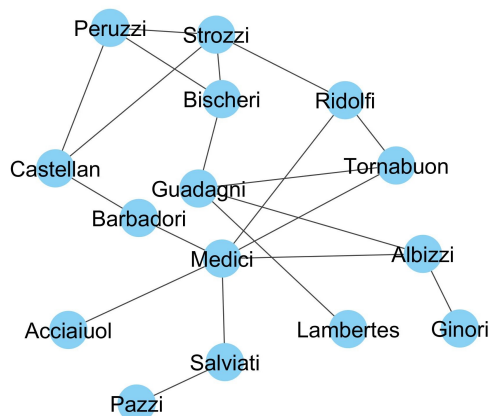


Figure 3.1: Padget’s Florentine families marriages relations graph [14]. Visualization done with Cytoscape[15].

Before continuing with the explanation of the centrality methods which we will describe in this chapter let us first introduce some rigor and limit the ambiguity of the notion of *centrality* by defining it very generally.

Definition 3.0.1 (Centrality). *Let $G = (V, E)$ be a graph, we call centrality any function f such that $f : V \rightarrow \mathbb{R}$.*

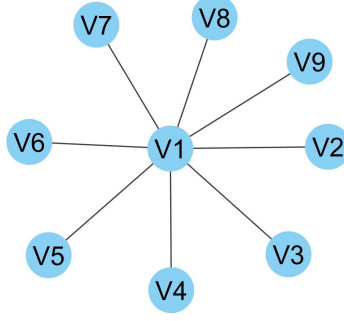


Figure 3.2: A star graph with 9 vertices.

3.1 Degree Centrality

Degree centrality is one of the most intuitive methods for introducing order to the nodes of a graph. It was introduced by Shaw in 1954 as a natural way of distinguishing central nodes based solely on their degree [8].

Definition 3.1.1 (Degree Centrality). *Let $G = (V, E)$ be a graph and v one of its nodes, the degree centrality of v is denoted by $C_D(v)$ and defined as $C_D(v) = \deg(v)$. In the case when $G = (V, E)$ is a directed graph the degree centrality is defined as $C_D(v) = \deg^-(v) + \deg^+(v)$*

In directed graphs we distinguish between two types of degree centralities which are *in-degree centrality* and *out-degree centrality*.

Definition 3.1.2 (In-Degree Centrality). *Let $G' = (V', E')$ be a directed graph and v one of its nodes, the in-degree centrality of v is denoted by $C_D^-(v)$ and defined as $C_D^-(v) = \deg^-(v)$.*

Definition 3.1.3 (Out-Degree Centrality). *Let $G' = (V', E')$ be a directed graph and v one of its nodes, the out-degree centrality of v is denoted by $C_D^+(v)$ and defined as $C_D^+(v) = \deg^+(v)$.*

If A and A' are the adjacency matrices of an undirected and directed graph respectively and a_{ij} , a'_{ij} their elements in the i -th row and j -th column accordingly, then the previously defined terms can be calculated by the following formulas:

$$C_D(v) = \sum_{i=1}^k a_{v,i},$$

$$C_D(v) = \sum_{i=1}^k a'_{i,v} + \sum_{i=1}^k a'_{v,i},$$

$$C_D^-(v) = \sum_{i=1}^k a'_{i,v},$$

$$C_D^+(v) = \sum_{i=1}^k a'_{v,i}.$$

As follows from the definitions $C_D(v)$ is large when the node v is adjacent to a high number of nodes and low when it does not have many direct connections. As extreme cases we consider the scenarios when $C_D(v) = K - 1$ and $C_D(v) = 0$ which are the maximal and minimal values that degree centrality can take in the case of a K -nodes graph. In the later case the node is completely disconnected from all the other nodes of the network (i.e., isolated node) whereas in the former case the node provides edges to all the other existing nodes in the graph. In the case of information transmission a node with high degree centrality is more knowledgeable of the data that is going through the network and its activity influences the network more than any other node. As a result such a node is considered as a major player of signal transmission, in that it allows direct communication with many nodes and lets those nodes to almost directly interact with each other through it. On the other hand a node with low degree centrality is more passive in the network and it is not directly involved in major information transfers because its position restricts it from direct communication with most other nodes. Considering that their position doesn't favor immediate visibility of other nodes in the network they are seen as peripheral nodes.

Since degree centrality is partly a function of the size of the network, the score of a node can be greater in cases of large graphs (i.e., having many nodes and edges) [16]. In order to capture the significance of the degree centrality with respect to the size of the graph, *normalized degree centrality* is used.

Definition 3.1.4 (Normalized Degree Centrality). *Let $G = (V, E)$ be a graph and let v be one of its nodes, the normalized degree centrality of v is denoted by $\tilde{C}_D(v)$ and defined as $\tilde{C}_D(v) = \frac{C_D(v)}{|V|-1}$.*

$C'_D(v)$ expresses the proportion of nodes directly connected to a specific node. Considering the minimal and maximal values of $C_D(v)$ it easily follows that $0 \leq C'_D(v) \leq 1$ where extreme values are reached in the cases of an isolated node (i.e.,

$C_D(v) = 0$) or a fully connected node (i.e., $C_D(v) = K - 1$). One can also think of it as the probability that a node v is connected to any other node given no other information regarding the network except for $C'_D(v)$.

So far we treated degree centrality as a measure of counting the number of nodes in the node's immediate neighborhood or the so called 'first-order zone' [16] (i.e., nodes that are directly connected via one edge with the node taken in consideration).

The concept of the *first-order zone* can be extended and generalized to the set of nodes which distance from a current node v is at most N , this is also referred to as the *Nth-order zone* [16] of the node.

Definition 3.1.5 (Nth-Order Zone). *Let $G = (V, E)$ be a graph and v one of its vertices, the Nth-order zone of v is denoted as $Z_n(v)$ and defined in the following way: $Z_n(v) = \{u \in V \mid d(v, u) \leq N\}$.*

If we denote by $C_{D_N}(v)$ the number of nodes in the Nth-order zone of v in a graph G consisting of K nodes then $C_{D_N}(v)$ can be computed as follows:

$$C_{D_N}(v) = \sum_{i=1}^K I(d_{v,i} \leq N),$$

where $d_{v,i}$ is the distance between nodes v and i , and $I(d_{v,i} \leq N)$ is a defined as:

$$I(d_{v,i} \leq N) = \begin{cases} 1 & \text{if } d_{v,i} \leq N \\ 0 & \text{if } d_{v,i} > N. \end{cases}$$

As we can see for $N = 1$ we obtain the usual degree centrality, while in the case when N is equal to the diameter of the network we obtain the *reachability index* [16].

Definition 3.1.6 (Reachability Index). *Let $G = (V, E)$ be a graph and v one of its vertices, the reachability index of v is defined as the cardinality of the set $V' = \{v_1 \dots v_r\} \subset V$ of vertices which distances from v are non-infinite.*

The concept of degree centrality of a node can help with determining the centrality index of the whole network. The index should be low in sparse graphs where all the vertices have more or less the same centrality degree, and high in the case where there is clearly one dominating vertex with a higher centrality degree

compared to all the other vertices. This simple idea led Freeman [17] to define the network centrality index based on degree centrality.

Definition 3.1.7 (Degree Centrality Index). *Let \mathbb{G} be the set of all graphs with $|V|$ vertices and let $G \in \mathbb{G}$. $\forall G' \in \mathbb{G}$ we denote by $v_{G'}^*$ its vertex with the highest degree centrality score. The degree centrality index of G denoted by $I_D(G)$ is defined as $I_D(G) = \frac{\sum_{i=1}^{|V|} (C_D(v_{G'}^*) - C_D(v_i))}{\max_{\forall G' \in \mathbb{G}} (\sum_{i=1}^{|V|} (C_D(v_{G'}^*) - C_D(v_i)))}$.*

It is easy to prove that the denominator takes its highest value in the case when G' is a star - that is, $\max_{\forall G' \in \mathbb{G}} (\sum_{i=1}^{|V|} (C_D(v_{G'}^*) - C_D(v_i))) = (|V| - 1)(|V| - 2)$. So finally the formula becomes:

$$I_D(G) = \frac{\sum_{i=1}^{|V|} (C_D(v_{G'}^*) - C_D(v_i))}{(|V| - 1)(|V| - 2)}.$$

Evidently the last definition captures the previously described principle of network centrality and $1 \geq I_D(G) \geq 0$, where extreme values are reached if G is either a fully connected graph (i.e $I_D(G) = 0$) or a star graph (i.e $I_D(G) = 1$).

In weighted graphs the degree centrality is defined by taking into consideration the weights of the edges as well [18].

Definition 3.1.8 (Degree Centrality in weighted graphs). *Let $G = (V, E)$ be a weighted graph, v one of its vertices and $V' \subset V$ the set of all the vertices adjacent to v , $=$. Then the degree centrality of v denoted by $C_D(v)$ is defined as*

$$C_D(v) = \sum_{i=1}^{|V'|} w(v_i).$$

The time complexity of the algorithm for finding the degree centrality of all nodes in an unweighted graph is linearly dependent on the number of its vertices and edges, thus having a time complexity $O(|V| + |E|)$ [19]. In the case of a weighted graph the algorithm for calculating the degree centrality is very similar to the case of an unweighted graph and the time complexity does not change.

Depending on the context of the problem, degree centrality can be useful in identifying important nodes to study or nodes of the network that we want to discard. As illustrated in [20] if we are looking on a social network (e.g., Twitter)

for interests that different users have by exploring their connections with other users or pages, we might want to exclude nodes with very high degree centrality because they don't give valuable information and introduce noise in our search. On the other hand if we are interested in finding the most popular user in the network, then identifying a node with the highest degree centrality might be a good indication to answer the question. In biology it has been used to measure the significance of a protein assuming that there is a positive correlation between its degree and the danger of its removal [21]. Another study by Hann et al. [9] suggested that the degree connectivity of a protein was positively correlated with its significance by studying PPI networks of yeast, worm and fly. Despite its simplicity and its intuitive motivation of existence, degree centrality offers a very restricted information regarding the network by limiting the view only on the locality of each node. It occurs very often for nodes inside the network to have very similar degree centrality scores [22] and in that case other centrality methods are required to make the difference.

Example 3.1.1 (Degree Centrality in Padget's Florentine Families). *Here we are giving the degree centrality values corresponding to each Florentine family according to their marriage relations with each other.*

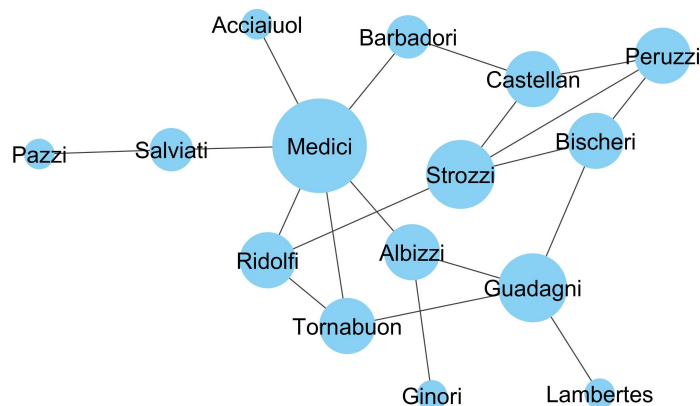


Figure 3.3: Padget's Florentine families marriages relations graph [14]. The relative sizes of the nodes correspond to their degree centrality. Visualization done with Cytoscape[15].

3.2 Closeness Centrality

Communication controllability over networks is one of the reasons why centrality methods have been studied. The ability to control the communication in a

Family	Wealth	Degree Centrality
Medici	103	12
Strozzi	146	8
Guadagni	8	8
Peruzzi	49	6
Castellan	20	6
Bischeri	44	6
Tornabuon	48	6
Albizzi	36	6
Ridolfi	27	6
Salviati	10	4
Barbadori	55	4
Pazzi	48	2
Lambertes	42	2
Ginori	32	2
Acciaiuol	10	2

Table 3.1: Padget’s Florentine families ordered by their corresponding degree centralities according to Figure 3.3.

network is an important property that a node can have, thus it is a reasonable argument to consider a node central based on the influence that it has on decentralizing the network, namely diminishing the potential of other nodes to control the flow. This principle motivated Bavelas in [23] to consider non-central those nodes which in order to connect with others have to transmit the signal through many intermediary nodes. Being able to communicate without depending on others introduced the concept of independence for a node and Leavit proposed that centrality should just measure the degree at which a node does not have to rely on indirect nodes in communication [8]. Both Bavelas and Leavit agreed that vertex independence is related to the closeness between it and all the other vertices. Many implications of the closeness property were made especially in social networks. The node with the highest closeness can transmit messages to the whole network in the most efficient way in terms of costs (e.g., time, space, money). The idea of determining central nodes those which occupy optimal positions that guarantee efficiency of signal transmission leaded to the definition of *closeness centrality* by Sabidussi [23].

Definition 3.2.1 (Closeness Centrality). *Let $G = (V, E)$ be a connected graph, $|V| > 1$, v one of its vertices and $d(v, v_i)$ the distance between vertices v and v_i .*

The closeness centrality of v denoted by $C_C(v)$ is defined as $C_C(v) = \frac{1}{\sum_{i=1}^{|V|} d(v, v_i)}$.

Notice that $1 > C_C(v) > 0$ since $\sum_{i=1}^{|V|} d(v, v_i) > 1$. According to the definition closeness centrality can be applied only to connected graphs otherwise if they have disconnected components the values of $C_C(v)$ would not produce informative results given that for each node there exists another one whose distance from the current is infinite. The size of the graph influences the value of $C_C(v)$ so in order to compare nodes belonging to graphs of different sizes Beauchamp proposed the *normalized closeness centrality* [8].

Definition 3.2.2 (Normalized Closeness Centrality). *Let $G = (V, E)$ be a connected graph, v one of its vertices and $d(v, v_i)$ the distance between vertices v and v_i . The normalized closeness centrality of v denoted by $\tilde{C}_C(v)$ is defined as*

$$\tilde{C}_C(v) = \frac{|V| - 1}{\sum_{i=1}^{|V|} d(v, v_i)}.$$

Unlike $C_C(v)$ in this case we have $1 \geq \tilde{C}_C(v) > 0$ and the upper bound is reached when v is adjacent to all the other vertices. Closeness centrality can be thought of as the inverse of the mean of the distances between a node and all the others. Since the minimum closeness centrality of a node in a connected graph $G(V, E)$ is $|V| - 1$ in the case of a node adjacent to all the others (e.g., center of a star, fully connected graphs), $\tilde{C}_C(v)$ can be understood as the inverse ratio by which the sum of the distances from a node to all the others surpasses its possible minimum. Both measures provide valuable information regarding the independence and efficiency of a vertex.

The algorithm for finding the closeness centrality requires the computation of the shortest paths from a given vertex to all the others which complexity is $O(|E|)$. Doing this procedure for all $|V|$ vertices will require $O(|V| \cdot |E|)$. This time complexity is very high considering the size of most real-life networks which is large. For this reason several approximate algorithms have been tried in order to compute $C_C(v)$ in feasible time. One of this approximate algorithms which uses heuristics was proposed by Branden and can run in $O(\alpha E)$, where α is between 10 and 100 [24]. This centrality metric is often preferable over degree centrality as it takes into consideration not only the nodes adjacent to a vertex but also other

nodes further away in the graph. As described by [19], nodes with high closeness centrality are appropriate locations for service facilities (facility location problem). In a study published in 2003 Wuchty and Stadler suggested a similarity between closeness centrality in biological networks and the facility location problem, it has also been used to discover important properties on the metabolic network of *E. coli*. [21].

The idea of closeness centrality can be extended to produce a *closeness centrality index* for the whole network. This index is a measure of homogeneity of distances in a graph and provides knowledge and means of comparisons between different networks.

Definition 3.2.3 (Closeness Centrality Index). *Let \mathbb{G} be the set of all connected graphs with $|V|$ vertices and let $G \in \mathbb{G}$. $\forall G' \in \mathbb{G}$ we denote by $v_{G'}^*$ its vertex with the highest closeness centrality score. The closeness centrality index of G denoted by $I_C(G)$ is defined as*

$$I_C(G) = \frac{\sum_{i=1}^{|V|} (\tilde{C}_C(v_G^*) - \tilde{C}_C(v_i))}{\max_{\forall G' \in \mathbb{G}} (\sum_{i=1}^{|V|} (\tilde{C}_c(v_{G'}^*) - \tilde{C}_c(v_i)))}.$$

Obviously $1 \geq I_C(G) \geq 0$ in any connected graph.

Theorem 3.2.1. $\sum_{i=1}^{|V|} (\tilde{C}_c(v_{G'}^*) - \tilde{C}_c(v_i))$ reaches its maximum when $v_{G'}^*$ is the center of a star.

Proof. Let's start from a star graph G' and evaluate the possible options of modifying it by analyzing how each modification will influence the value of the expression

$$\sum_{i=1}^{|V|} (\tilde{C}_c(v_{G'}^*) - \tilde{C}_c(v_i)):$$

1. Adding an edge between two non-central nodes will keep the value of $\sum_{i=1}^{|V|} \tilde{C}_c(v_{G'}^*)$ unchanged but will increase $\sum_{i=1}^{|V|} \tilde{C}_c(v_i)$, thus decreasing the value of $\sum_{i=1}^{|V|} (\tilde{C}_c(v_{G'}^*) - \tilde{C}_c(v_i))$.
2. Deleting an edge is forbidden because it will disconnect the graph making closeness centrality undefined.

3. Switching an edge from the center node and putting it between two other nodes by keeping the graph connected will reduce $\sum_{i=1}^{|V|} \tilde{C}_c(v_{G'}^*)$ and increase $\sum_{i=1}^{|V|} (\tilde{C}_c(v_{G'}^*) - \tilde{C}_c(v_i))$ by making the graph sub-optimal with respect to the expression $\sum_{i=1}^{|V|} (\tilde{C}_c(v_{G'}^*) - \tilde{C}_c(v_i))$.

□

In the case of a star graph we have:

$$\sum_{i=1}^{|V|} (\tilde{C}_c(v_{G'}^*) - \tilde{C}_c(v_i)) = (|V| - 1) \left(\frac{|V| - 1}{|V| - 1} - \frac{|V| - 1}{2(|V| - 2) + 1} \right) = \frac{|V|^2 - 3|V| + 2}{2|V| - 3},$$

and finally:

$$I_C(G) = \frac{\sum_{i=1}^{|V|} (\tilde{C}_C(v_G^*) - \tilde{C}_C(v_i))}{(|V|^2 - 3|V| + 2)/(2|V| - 3)}.$$

Example 3.2.1 (Facility Location). *A new village is being built for the homeless families whose houses were destroyed by a devastating earthquake. Suppose that a graph G is given which represents the positions where new buildings will be placed and the roads that will connect them. Find the most favorable position in which the public service buildings (e.g., health center, school, church, mosque) should be located.*

As stated previously placing the public services in vertices that have higher closeness centrality an excluding vertices that represent road intersection would provide the best logistics in order for people to easily access them.

Example 3.2.2 (Closeness Centrality in Padgett's Florentine Families). *Here we are giving the closeness centrality values corresponding to each Florentine family according to their marriage relations with each other.*

Example 3.2.3. *A given graph represents similarities between DNA strands such that two nodes are connected only if they share similarity above a given threshold. Find the DNA sharing most similarities with all the others.*

The node with highest similarity is the one from which the sum of distances from all the others is the smallest, hence the one having highest closeness centrality.

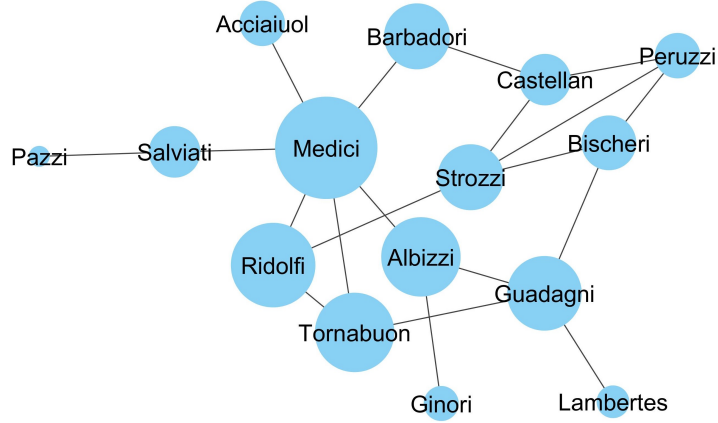


Figure 3.4: Padget's Florentine families marriages relations graph [14]. The relative sizes of the nodes correspond to their closeness centrality. Visualization done with Cytoscape[15].

Family	Wealth	Closeness Centrality
Medici	103	0.56
Ridolfi	27	0.5
Tornabuoni	48	0.48275862
Albizzi	36	0.48275862
Guadagni	8	0.46666667
Strozzi	146	0.4375
Barbadori	55	0.4375
Bischeri	44	0.4
Castellani	20	0.38888889
Salviati	10	0.38888889
Peruzzi	49	0.36842105
Acciaiuoli	10	0.36842105
Ginori	32	0.33333333
Lambertes	42	0.3255814
Pazzi	48	0.28571429

Table 3.2: Padget's Florentine families ordered by their corresponding closeness centralities according to Figure 3.4.

3.3 Betweenness Centrality

The frequency of occurrence that a vertex is in many of the shortest paths between all pairs of other nodes can also be seen as a centrality measure. For example in a star graph it is not possible to go from one node to the other without traversing the center node. This measure gives again a special position to the center of the star. Bavelas and Shaw in their studies on social centrality suggested that

a person who is able to connect other pairs in the network which cannot reach each other in a shorter way should be considered important and should be given higher centrality score because they have responsibility in the healthy transferal of information throughout the network [8]. The amount of control that a vertex has in the information exchanged between other vertices via their communications paths is measured by *betweenness centrality*.

Definition 3.3.1 (Betweenness Centrality). *Let $G = (V, E)$ be a graph and v one of its vertices. Given that $g_{uw}(v)$ represents the number of occurrences of v in the shortest paths between a pair of nodes u and w , and g_{uw} is the total number of shortest paths connecting u with w . Then betweenness centrality of v is denoted by $C_B(v)$ and defined as $C_B(v) = \sum_{\substack{u < w \\ u \neq v \neq w}} \frac{g_{uw}(v)}{g_{uw}}$.*

It is easy to calculate $C_B(v)$ when there are only single shortest paths connecting pairs of nodes in the network. However the situation becomes more complex when there are multiple shortest paths. Methods for calculating $C_B(v)$ have been specified by Harary et al. and they can be easily implemented in computer programs using matrices [8]. Depending on the number of shortest paths between two vertices that contain a vertex v and the total number of shortest paths between them, we can distinguish among full and partial control of v over their connection. Thus, if v belongs to all the shortest paths between the pair, we say that it has full control, whereas if there exists one such path that it does not belong to, we say that its control over the communication of the pair is partial. In the case of partial control the overall centrality decreases. Betweenness centrality of v over one pair of nodes can also be interpreted in terms of probability.

Example 3.3.1. *Suppose we have a graph $G = (V, E)$ and let $u, w, v \in V$. What is the potential of v to control the information which is communicated between u and w ?*

This potential of v is equal to the probability that it has to belong to a randomly selected shortest path connecting u and w which is $\frac{g_{uw}(v)}{g_{uw}}$.

Similar to the other measures considered so far, this centrality is also a function of the size of the graph and in order to obtain a standardized score with which comparisons among nodes belonging to different graphs can be made *normalized betweenness centrality* is used.

Definition 3.3.2 (Normalized Betweenness Centrality). *Let $G = (V, E)$ be a graph and let v be one of its vertices, the normalized betweenness centrality of v , denoted by $\tilde{C}_B(v)$ is calculated as $\tilde{C}_B(v) = \frac{2C_B(v)}{|V|^2 - 3|V| + 2}$.*

$\tilde{C}_B(v)$ is in fact the ratio between $C_B(v)$ and the maximum value of betweenness centrality that a vertex in G can have. Considering that $C_B(v) = \sum_{\substack{u < w \\ u \neq v \neq w}} \frac{g_{uw}(v)}{g_{uw}}$

it is easy to observe that the maximum is reached when $\frac{g_{uw}(v)}{g_{uw}} = 1$ because $g_{uw}(v) \leq g_{uw}$ and its highest value will be 1. As a result we will have:

$$\max(C_B(v)) = \max\left(\sum_{\substack{u < w \\ u \neq v \neq w}} \frac{g_{uw}(v)}{g_{uw}}\right) = \binom{K-1}{2} = \frac{|V|^2 - 3|V| + 2}{2}.$$

Theorem 3.3.1. *The necessary and sufficient condition for a graph $G = (V, E)$ to be a star is for a node v to exist such that: $C_B(v) = \frac{|V|^2 - 3|V| + 2}{2}$.*

Proof. Let $G = (V, E)$ be a star graph and v its central node, then it is clear that $C_B(v) = \frac{|V|^2 - 3|V| + 2}{2}$. On the other hand let's denote by $S_{u,w}$ the set consisting of the sets of nodes included in the shortest paths between u and w , if $G = (V, E)$ is not a star then for every vertex $v \in V$ there will exist vertices u and w such that for all $P \in S_{u,w}$, we have $v \notin P$ (i.e., $g_{uw}(v) = 0$). Thus,

$$C_B(v) \leq \max\left(\sum_{\substack{u < w \\ u \neq v \neq w}} \frac{g_{uw}(v)}{g_{uw}}\right) - 1 = \frac{|V|^2 - 3|V| + 2}{2} - 1.$$

□

Betweenness centrality can also be used to compute the centrality index of a graph. This index was introduced by Freeman [25] and is often applied when ranking graphs according to their centrality distributions.

Definition 3.3.3 (Betweenness Centrality Index). *Let \mathbb{G} be the set of all graphs with $|V|$ vertices and let $G \in \mathbb{G}$. $\forall G' \in \mathbb{G}$ we denote by $v_{G'}^*$ its vertex with the highest betweenness centrality score. The betweenness centrality index of G denoted by $I_B(G)$ is defined as*

$$I_B(G) = \frac{\sum_{i=1}^{|V|} (\tilde{C}_B(v_G^*) - \tilde{C}_B(v_i))}{\max_{\forall G' \in \mathbb{G}} \left(\sum_{i=1}^{|V|} (\tilde{C}_c(v_{G'}^*) - \tilde{C}_c(v_i)) \right)}.$$

It is clear that the value of $\sum_{i=1}^{|V|} (\tilde{C}_c(v_{G'}^*) - \tilde{C}_c(v_i))$ is maximal in the case when the term $\sum_{i=1}^{|V|} \tilde{C}_c(v_{G'}^*)$ has the highest value and the term $\sum_{\substack{i=1 \\ v_i \neq v_{G'}^*}}^{|V|} \tilde{C}_c(v_i)$ has the lowest value in any graph of $|V|$ vertices. The previous conditions are satisfied when G' is a star. Thus we have:

$$\max_{\forall G' \in \mathbb{G}} \left(\sum_{i=1}^{|V|} (\tilde{C}_c(v_{G'}^*) - \tilde{C}_c(v_i)) \right) = \frac{2 \cdot (|V| - 1)}{(|V| - 1)(|V| - 2)} \cdot \frac{(|V| - 1)(|V| - 2)}{2} = |V| - 1.$$

Elaborating a bit further the definition of $I_B(G)$ it is possible to express it in terms of $C_B(v_{G'}^*)$ and $C_B(v)$ as follows:

$$\begin{aligned} I_B(G) &= \frac{\sum_{i=1}^{|V|} (\tilde{C}_B(v_{G'}^*) - \tilde{C}_B(v_i))}{|V| - 1} = \frac{2 \sum_{i=1}^{|V|} \frac{C_B(v_{G'}^*) - C_B(v_i)}{|V|^2 - 3|V| + 2}}{|V| - 1} \quad (\text{Definition 3.3.2}) \\ &= \frac{2 \sum_{i=1}^{|V|} (C_B(v_{G'}^*) - C_B(v_i))}{|V|^3 - 4|V|^2 + 5|V| - 2}. \end{aligned}$$

The values of $I_B(G)$ range between 0 and 1. It is 0 in the case when the graphs' nodes have equal $C_B(v)$ and 1 in the case of the star or wheel graphs [25]. Nodes having high $C_B(v)$ are considered important in terms of information controllability, since in order for pairs of nodes to communicate with each other efficiently the signal will most probably go through those nodes. A node with high betweenness centrality has a significant implication in the graph's connectivity because its removal may imply the disconnection of the graph. Measuring betweenness score of vertices in large biological networks for PPI reveals important information on their potential involvement in diseases. Sahoo et al. [26] suggested that disturbing these type of nodes may cause sensitive topological changes in PPI networks and hence they require particular attention when differentiating between cancer and normal PPI through network analysis.

There are many algorithms used to calculate the betweenness centrality and their computational complexity is a trade off between time and memory space. Perhaps one of the most popular is Brandes' algorithm [24] with a time complexity of $O(|E| \cdot |V|)$. Recently Nasre et al. came with a better time efficient algorithm

which has both time and space complexity of $O(\nu^*|V|)$ where ν^* is the maximum number of edges that lie on the shortest paths of every vertex of the graph [27].

Example 3.3.2 (Betweenness Centrality in Padget’s Florentine Families). *Here we are giving the betweenness centrality values corresponding to each Florentine family according to their marriage relations with each other.*

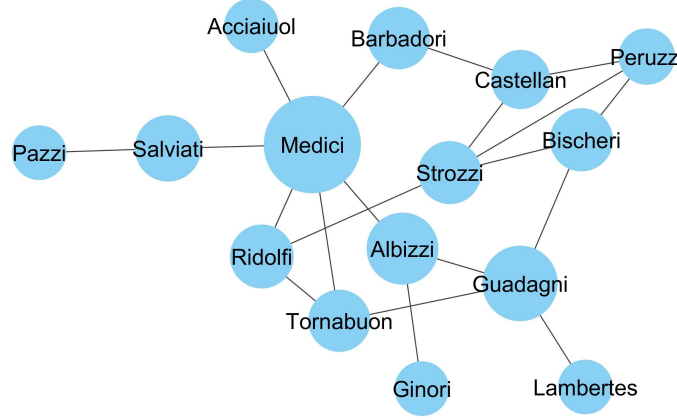


Figure 3.5: Padget’s Florentine families marriages relations graph [14]. The relative sizes of the nodes correspond to their betweenness centrality. Visualization done with Cytoscape[15].

Family	Wealth	Betweenness Centrality
Medici	103	0.52197802
Guadagni	8	0.25457875
Albizzi	36	0.21245421
Salviati	10	0.14285714
Ridolfi	27	0.11355311
Bischeri	44	0.1043956
Strozzi	146	0.1025641
Barbadori	55	0.09340659
Tornabuon	48	0.09157509
Castellan	20	0.05494505
Peruzzi	49	0.02197802
Acciaiuol	10	0.0
Ginori	32	0.0
Lambertes	42	0.0
Pazzi	48	0.0

Table 3.3: Padget’s Florentine families ordered by their corresponding betweenness centralities according to Figure 3.5.

3.4 Harmonic Centrality

As was previously discussed closeness centrality comes with a major drawback in the case when the network is disconnected because the distances between vertices belonging to separate components do not exist or are considered to be infinite. In that case for each vertex v of the graph we would have $C_C(v) = \frac{1}{\dots+\infty+\dots} = 0$ which is a trivial metric. One way to solve this problem is by including in the computation only the closeness centrality of the vertices found in the largest component. However, the former trick does not help when there are many large components. It is possible to compute the degree and betweenness centralities of any node regardless of the graph type to which they belong to (i.e., either connected or disconnected). For this reason, multiple ways have been tried to overcome this limitation of closeness centrality. Csardi et al. [28] proposed the substitution of infinite distances with the length of the longest path that can exist between the vertices in a network with K nodes (i.e., $K - 1$).

Definition 3.4.1 (Closeness Centrality for disconnected graphs). *Let $G = (V, E)$ be a graph, v one of its vertices and $d(v, v_i)$ the distance between v and v_i . The closeness centrality of v belonging to one of its separate components denoted by $C_\alpha(v)$ is defined as $C_\alpha(v) = \frac{|V|-1}{\sum_{\substack{i=1 \\ v \neq v_i}}^{|V|-1} d(v, v_i) + m\alpha}$, where m is the number of vertices not belonging to the component of v and $\alpha \in \mathbb{R}_+$ is a constant such that $\alpha \geq \text{diam}(G)$.*

One other method for handling proximity in disconnected graphs created by Rochat and called *harmonic centrality* was based on the *mean distance* as defined by Newman [29].

Definition 3.4.2 (Mean Distance). *Let $G = (V, E)$ be a graph. The mean distance is denoted by $l(G)$ and defined as $l(G) = \frac{2}{|V|^2+|V|} \sum_{i \neq j} \frac{1}{d(v_j, v_i)}$.*

Definition 3.4.3 (Harmonic Centrality). *Let $G = (V, E)$ be a graph, v one of its vertices and $d(v, v_i)$ the distance between v and v_i , its harmonic centrality is denoted by $C_H(v)$ and defined as $C_H(v) = \sum_{\substack{i=1 \\ v \neq v_i}}^{|V|-1} \frac{1}{d(v, v_i)}$.*

Harmonic centrality is a recently introduced method for analyzing networks and there were many researchers promoting it on the same time. Boldi et al. [30] were motivated by Marchiori and Latora who proposed the usage of harmonic

mean instead of the arithmetic mean of distances to compute closeness centrality in the case of directed networks. Saramaki et al. [31] used it as a solution for dealing with closeness centrality in disconnected networks. Cohen et al. [32] used a method called spatially-decaying aggregate which is a generalization of harmonic centrality. However, in this thesis we consider it as defined by Rochat in [28].

The normalized version of $C_H(v)$ is obtained by dividing it with the maximal value that harmonic centrality can reach which is $|V| - 1$ if v is the center in a star graph:

$$\tilde{C}_H(v) = \frac{1}{|V| - 1} \sum_{\substack{i=1 \\ v \neq v_i}}^{|V|-1} \frac{1}{d(v, v_i)}.$$

It is easy to observe that harmonic centrality increases as the number of vertices belonging to the component becomes larger, thus giving higher values to nodes in massive components. Another comment is that it gives lower scores to nodes in disconnected graphs compared to connected ones, thus reflecting the inability to communicate between separate components [28].

As we previously saw in sections 3.1, 3.2, 3.3 it is possible to modify nodes' centralities in order to produce a representative index for the whole graph. The same thing remains true in the case of harmonic centrality thus we define the *harmonic centrality index* of a graph.

Definition 3.4.4 (Harmonic Centrality Index). *Let \mathbb{G} be the set of all graphs with $|V|$ vertices and let $G \in \mathbb{G}$. $\forall G' \in \mathbb{G}$ we denote by $v_{G'}^*$ its vertex with the highest harmonic centrality score. The harmonic centrality index of G denoted by*

$$I_H(G) \text{ is defined as } I_H(G) = \frac{\sum_{i=1}^{|V|} (\tilde{C}_H(v_G^*) - \tilde{C}_H(v_i))}{\max(\sum_{i=1}^{|V|} (\tilde{C}_H(v_{G'}^*) - \tilde{C}_H(v_i)))}.$$

The maximal value of $\sum_{i=1}^{|V|} (\tilde{C}_H(v_{G'}^*) - \tilde{C}_H(v_i))$ is achieved by studying the case of a star graph in which we will have:

$$\begin{aligned} \sum_{i=1}^{|V|} (\tilde{C}_H(v_{G'}^*) - \tilde{C}_H(v_i)) &= \frac{1}{|V| - 1} \sum_{i=1}^{|V|} (C_H(v_{G'}^*) - C_H(v_i)) \\ &= \frac{1}{|V| - 1} (|V| - 1)(|V| - 1 - \lfloor \frac{1}{2}(|V| - 2) + 1 \rfloor) \\ &= \frac{|V| - 2}{2}. \end{aligned}$$

Finally $I_H(G)$ can be rewritten in the form:

$$I_H(G) = \frac{2}{(|V| - 2)(|V| - 1)} \sum_{i=1}^{|V|} (C_H(v_G^*) - C_H(v_i)).$$

As we have seen so far there is no fundamental difference between the procedures of finding closeness and harmonic centralities, hence the amount of computation that we have to do in order to extract the needed information from the graph is the same. Finding $C_H(v)$ of a node $v \in V$ from $G = (V, E)$ requires computing the shortest paths from v to all the nodes in V , and finding the highest v requires applying the procedure for all $v \in V$. The total computational complexity for calculating the harmonic centrality in a network $G = (V, E)$ is $O(|V| \cdot |E|)$ as it is described by Rochat [28]. A more efficient method to determine harmonic centrality via approximate algorithms is proposed in [33].

Example 3.4.1 (Harmonic Centrality in Padget’s Florentine Families). *Here we are giving the harmonic centrality values corresponding to each Florentine family according to their marriage relations with each other.*

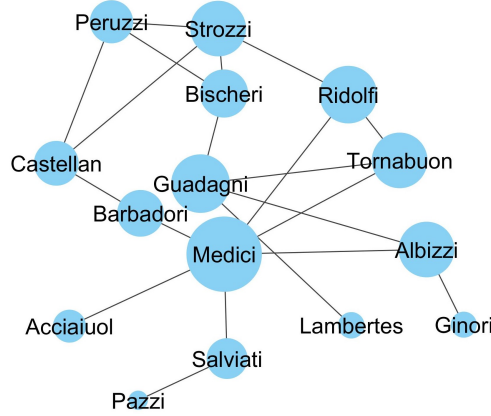


Figure 3.6: Padget’s Florentine families marriages relations graph [14]. The relative sizes of the nodes correspond to their harmonic centrality. Visualization done with Cytoscape[15].

3.5 Degree Prestige

In the analysis of directed networks the in-degree centrality of a node v is often referred to as the *degree prestige* of that node and it is denoted by $P_D(v)$. Degree prestige is considered an important property based on the assumption that nodes

Family	Wealth	Harmonic Centrality
Medici	103	9.5000000000000002
Guadagni	8	8.083333333333332
Ridolfi	27	7.999999999999999
Albizzi	36	7.833333333333332
Strozzi	146	7.833333333333332
Tornabuon	48	7.833333333333332
Bischeri	44	7.199999999999999
Barbadori	55	7.083333333333332
Castellan	20	6.916666666666665
Peruzzi	49	6.783333333333332
Salviati	10	6.583333333333332
Acciaiuol	10	5.916666666666666
Lambertes	42	5.366666666666667
Ginori	32	5.333333333333334
Pazzi	48	4.766666666666667

Table 3.4: Padget’s Florentine families ordered by their corresponding harmonic centralities according to Figure 3.6.

which are pointed to by directed edges coming from other nodes (namely nominated by those nodes) have higher importance in the network. Degree prestige is normalized to a value between 0 and 1 by dividing with the maximum value of in-degree that it can have. This number is $K - 1$ in the case of a K -vertices network (i.e., node v is nominated by all the other $K - 1$ nodes of the graph), hence its normalized version becomes:

$$\tilde{P}_D(v) = \frac{P_D(v)}{K - 1}.$$

This method suffers from the same limitations as degree centrality. The information that it can give is restricted and localized since it relies only on direct connections (i.e., nodes one edge distant from the actual node) and disregards the prestige of those nodes, thus not taking into consideration overall aspects of the network.

Example 3.5.1 (Degree Prestige in Padget’s Florentine Families). *Here we are giving the degree prestige values corresponding to each Florentine family according to their marriage relations with each other.*

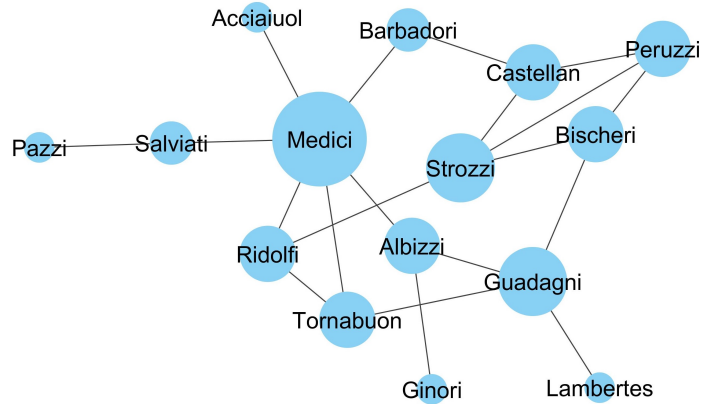


Figure 3.7: Padget’s Florentine families marriages relations graph [14]. The relative sizes of the nodes correspond to their degree prestige. Visualization done with Cytoscape[15].

Family	Wealth	Degree Prestige
Medici	103	6
Strozzi	146	4
Guadagni	8	4
Peruzzi	49	3
Castellan	20	3
Bischeri	44	3
Tornabuon	48	3
Albizzi	36	3
Ridolfi	27	3
Salviati	10	2
Barbadori	55	2
Pazzi	48	1
Lambertes	42	1
Ginori	32	1
Acciaiuol	10	1

Table 3.5: Padget’s Florentine families ordered by their corresponding degree prestige according to Figure 3.7.

3.6 Eigenvector-based Prestige

As opposed to degree prestige which restricts the prestige of the node only to its locality (i.e., it only measures the nodes directly connected to the node without considering their prestige), *eigenvector-based prestige* captures also the popularity (i.e., prestige, in-degree) of the other nodes and determines how much they contribute to each other and to the node which we are interested in. As a result

it gives a more complete description of the node and its centrality by considering it in a global perspective (i.e., by examining all the nodes of the network, not just those in its immediate neighborhood). Philip Bonacich suggested that the centrality of a vertex in a graph should be proportionally related with the centrality of its neighbors - that is, higher centrality of neighbors should induce higher centrality for the vertex taken into consideration [34]. Mathematically the idea is equivalent to solving a system of linear equations of the form:

$$\begin{aligned} P(v_1) &= a_{1,1}P(v_1) + a_{1,2}P(v_2) + \dots + a_{1,n}P(v_n), \\ &\dots \\ P(v_n) &= a_{n,1}P(v_1) + a_{n,2}P(v_2) + \dots + a_{n,n}P(v_n). \end{aligned}$$

Here $a_{i,j}$ is an element of the adjacency matrix of the network, as such its values can be either 1 or 0 and that guarantees the influence of only those nodes directly connected to a vertex when calculating its centrality. Based on this idea the following definition emerged:

Definition 3.6.1 (Eigenvector Centrality of a graph). *Let $G = (V, E)$ be a strongly connected graph and let A be the adjacency matrix representation of G . We define \vec{P} to be the eigenvector centrality of the vertices of G and compute it by solving the following equation: $\lambda \vec{P} = A \vec{P}$, where λ is the largest eigenvalue in absolute value that can satisfy the previous equation.*

For the sake of completeness and in order to be consistent with the initial Definition 3.0.1 for centrality we also define eigenvector centrality in the context of a vertex.

Definition 3.6.2 (Eigenvector Centrality of a node). *Let $G = (V, E)$ be a strongly connected graph and let $\vec{P} = (p_1, p_2, \dots, p_{|V|})^T$ be its eigenvector centrality for $v, v_2, \dots, v_{|V|}$ respectively. We call $P(v_i)$ the eigenvector centrality of vertex v_i and compute it as $P(v_i) = p_i$.*

Alternatively the last definition can also be found in the following form as described by [35]:

$$P(u) = \frac{1}{\lambda} \sum_v A_{v,u} P(v).$$

It is important to notice here that even though there are multiple values of λ which satisfy the equation we can only choose the maximal value in the set of

solutions because we want the components $p_1, p_2, \dots, p_{|V|}$ of \vec{P} to be positive and this is guaranteed by an implication of the Pierron-Frobenius theorem given that A as an adjacency matrix is squared and all its elements are positive [36].

In order to capture the basic intuition behind eigenvector centrality let's consider a simple example of a network and how we can rank its nodes based on that principle.

Example 3.6.1. *Rank the nodes of the network showed in Figure 3.8 based not only on their individual centrality degree but also by considering the degree of their neighbors.*

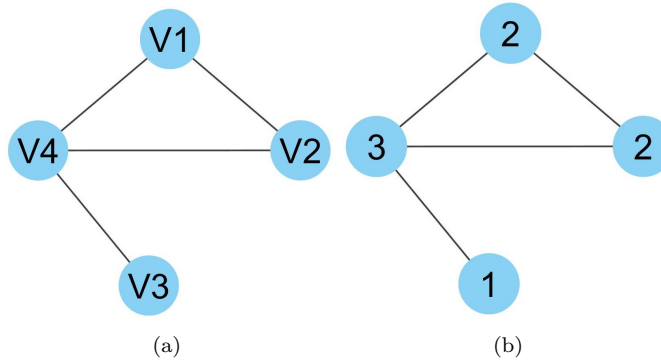


Figure 3.8: The same graph G with vertices labeled in different ways. (a) Graph G with vertices labeled $V1, V2, V3, V4$ according to the order that they appear in the adjacency matrix.(b) Graph G with vertices labeled according to their degree centrality.

The adjacency matrix of G will be $A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ and let $\vec{d} = \begin{pmatrix} 2 \\ 2 \\ 3 \\ 1 \end{pmatrix}$ be a

vector representing the centrality degree of vertices $V1, V2, V3, V4$ respectively. Let's now consider the following product

$$A\vec{d} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 2 \\ 3 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \times 2 + 1 \times 2 + 1 \times 3 + 0 \times 1 \\ 1 \times 2 + 0 \times 2 + 1 \times 3 + 0 \times 1 \\ 1 \times 2 + 1 \times 2 + 0 \times 3 + 1 \times 3 \\ 0 \times 2 + 0 \times 2 + 1 \times 3 + 0 \times 1 \end{pmatrix} = \begin{pmatrix} 5 \\ 5 \\ 7 \\ 3 \end{pmatrix}.$$

As we can see each of the components of the new vector adds something to the score of each node corresponding to the degree of their respective neighbors. Figure 3.9 shows the scores of the nodes of graph G after the multiplication.

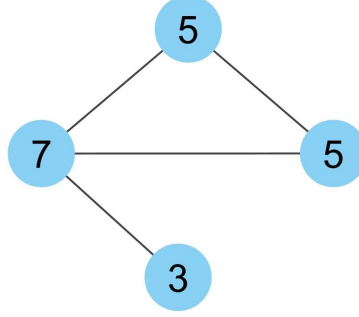


Figure 3.9: Graph G with vertices labeled according to their new scores.

We can again multiply the adjacency matrix A with the new vector that we obtained and we will get another vector which components will be the sum of the scores of the neighbors and assign those components as scores to the vertices. This process can be continued indefinitely and the components of the new vectors will expand after each iteration. What can be even more interesting is achieving a point of equilibrium in which every new vector will change by its predecessor only by a constant (i.e., the ratios of their respective components will be the same). This idea can be reduced into solving the equation on \vec{x} :

$$A\vec{x} = \lambda\vec{x}.$$

At this point it is clear that the equation is satisfied for \vec{x} and λ being the eigenvector and eigenvalue of the matrix A . In order to answer to the initial

ranking problem we just have to solve the equation: $\begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \vec{x} = \lambda\vec{x}$ for

λ being maximal and find that $\vec{x} = (\frac{\sqrt{5}+1}{2}, \frac{\sqrt{5}+1}{2}, \frac{\sqrt{5}+3}{2}, 1)^T$ and $\lambda = \frac{\sqrt{5}+3}{2}$ are the solutions, hence the final ranking will be $V3, V1, V2, V4$. In order to always be able to find a solution to this type of equations it is important for the adjacency matrix to correspond to a strongly connected graph and no loops should be present.

Example 3.6.2 (Eigenvector Centrality in Padget's Florentine Families). Here we are giving the eigenvector centrality values corresponding to each Florentine family according to their marriage relations with each other.

The time complexity of eigenvector based centrality is inherently connected to the time complexity of the algorithms for matrix multiplication which is $O(|V|^3)$.

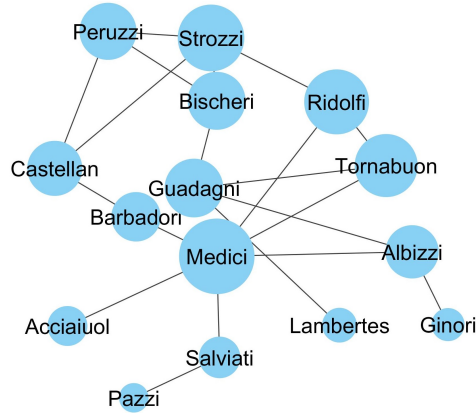


Figure 3.10: Padget’s Florentine families marriages relations graph [14]. The relative sizes of the nodes correspond to their eigenvector centrality. Visualization done with Cytoscape[15].

Family	Wealth	Eigenvector Centrality
Medici	103	0.4303154258349923
Strozzi	146	0.3559730326460451
Ridolfi	27	0.3415544259074365
Tornabuon	48	0.325846704169574
Guadagni	8	0.28911715732265014
Bischeri	44	0.2827943958713356
Peruzzi	49	0.2757224374104833
Castellan	20	0.25902003784235145
Albizzi	36	0.2439605296754477
Barbadori	55	0.21170574706479847
Salviati	10	0.14592084164171834
Acciaiuol	10	0.1321573195285342
Lambertes	42	0.08879253113499551
Ginori	32	0.0749245316027793
Pazzi	48	0.044814939703863084

Table 3.6: Padget’s Florentine families ordered by their corresponding eigenvector centralities according to Figure 3.10.

In a comparison study among 6 different centrality methods eigenvector based prestige was proven to be very important in identifying crucial proteins when analyzing PPI network of yeast [37]. The only method which outperformed its results was sub-graph centrality which is another method belonging to the group of spectral centralities [38].

3.7 Katz Prestige

Katz prestige is a measurement of centrality similar to eigenvector-based prestige, proposed by Leo Katz [39]. This method scores a node by taking into consideration all the vertices of the network as opposed to ranking it in its neighborhood. Katz centrality aims to rank a node not only by considering the prestige of the nodes to which it is directly connected but also considering the number of indirect connections. It does so by expressing the score function of a node in terms not only of the first order connections but generally in terms of n th order connections. Both direct and indirect connections count in increasing the score of the node but their contribution decreases with the increase of the connection chain. The score is calculated with the help of a power series.

Definition 3.7.1 (Katz Centrality). *Let $G = (V, E)$ be a loop-free graph and A its adjacency matrix. We define the Katz centrality of vertex $v_i \in V$ as follows:*

$$C_{Katz}(v_i) = \sum_{k=1}^{\infty} \sum_{j=1}^n \alpha^k (A^k)_{ji}.$$

The above formula perfectly captures the explanation of this centrality. We notice that $(A^k)_{ji}$ is the a_{ij} element of the k th power of the adjacency matrix which value expresses the number of paths of length k in G that connects vertex v_j with v_i and is given by Theorem 3.7.1.

Theorem 3.7.1. *Let $G = (V, E)$ be a graph and A its adjacency matrix. Then $(A^k)_{ij}$ is the number of paths of length k between v_i and v_j in G .*

The factor α on the other hand should be a non-negative value smaller to the inverse of the largest eigenvalue ($0 \leq \alpha < \frac{1}{\lambda_{max}}$). Doing some algebraic transformations on the definition we can give it a more compact form using matrix notations:

$$C_{Katz}(v_i) = \alpha \sum_{j=1}^{|V|} A_{ji} + \alpha^2 \sum_{j=1}^{|V|} A_{ji}^2 + \dots;$$

$$\vec{C}_{Katz} = (\alpha A^T + \alpha^2 (A^T)^2 + \alpha^3 (A^T)^3 + \dots) \vec{1} = \sum_{n=1}^{\infty} (\alpha^n (A^T)^n) \vec{1} = \left(\sum_{n=0}^{\infty} (\alpha A^T)^n - I \right) \vec{1};$$

$$\sum_{n=0}^{\infty} (\alpha A^T)^n = (I - \alpha A^T)^{-1};$$

$$\vec{C}_{Katz} = ((I - \alpha A^T)^{-1} - I)\vec{1}.$$

Here $\vec{1}$ is a vector with $|V|$ components equal to 1 and I is the identity matrix of size $|V| \times |V|$ with ones in the major diagonal (i.e., bottom right to top left) and zeros elsewhere. As we previously mentioned the factor α can be chosen between 0 and $\frac{1}{\lambda_{max}}$. The smaller we choose the factor the less significant will be the impact of indirect paths to the score of the node. A special case is when we choose $\alpha = 0$ and in that case we get the degree centrality of the node, this is why Katz centrality is considered a generalization of degree centrality. To emphasize the role of α let's consider an example which is created on the spirit of a solved exercise from [34].

Example 3.7.1. *Figure 3.11 represents a small network of proteins interacting with each other. We rank the proteins according to their importance by using Katz centrality and consider different values for α . The solution of the equation*

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \vec{x} = \lambda \vec{x},$$

suggests that $\lambda_{max} = 1$ which leaves us the choice of taking α from the half-open interval $[0, 1)$. For the purpose of the exercise we arbitrarily choose two distant alphas (e.g., $\alpha_1 = 0.1$ and $\alpha_2 = 0.9$) and calculate Katz centrality for each of the vertices.

Protein	$\alpha = 0.1$	$\alpha = 0.9$
P1	0.457	47.957
P2	0.266	46.563
P3	0.2	1.8
P6	0.1	0.9
P5	0.0	0.0
P4	0.0	0.0

Table 3.7: Katz index for each of the proteins in the network represented in Figure 3.11.

As the value of α increases from 0.1 to 0.9 the importance of indirect connections becomes more obvious in the scoring of the nodes. We can observe from Table 3.7 that the difference between $P2$ and $P3$ is quite minor for the lower α because they both receive two direct edges, but when indirect edges' weights start counting more (e.g., $\alpha = 0.9$) the difference among them prevails.

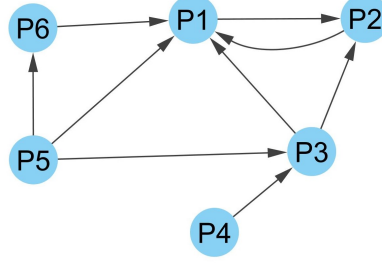


Figure 3.11: Interaction of proteins.

Example 3.7.2 (Katz Centrality in Padget's Florentine Families). *Here we are giving the Katz centrality values corresponding to each Florentine family according to their marriage relations with each other. We have selected $\alpha = 0.1$.*

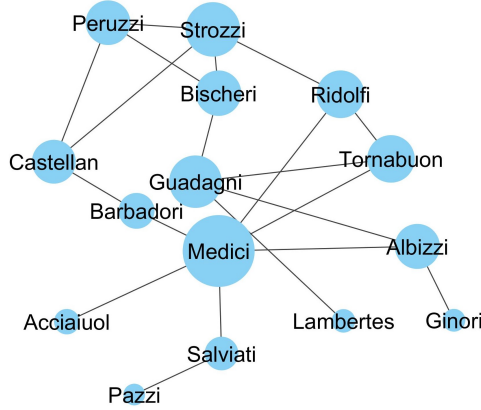


Figure 3.12: Padget's Florentine families marriages relations graph [14]. The relative sizes of the nodes corresponds to their Katz centrality. Visualization done with Cytoscape[15].

The time complexity for calculating Katz centrality is bounded by matrix multiplication which makes it $O(|V|^3)$ but approximate algorithms can be used with time complexity $O(|V| + |E|)$ [40].

Family	Wealth	Katz Centrality
Medici	103	0.3338248231609825
Strozzi	146	0.2898701598126901
Guadagni	8	0.2847793279616722
Ridolfi	27	0.2726911301187117
Tornabuon	48	0.27222832843835165
Bischeri	44	0.2670717598203384
Albizzi	36	0.2659283896264955
Peruzzi	49	0.26508006405086565
Castellan	20	0.2628706655772262
Barbadori	55	0.24276836030557358
Salviati	10	0.23716281096373137
Acciaiuol	10	0.21648130143376318
Lambertes	42	0.21157675618005325
Ginori	32	0.2096916639713015
Pazzi	48	0.20681510895407587

Table 3.8: Padget's Florentine families ordered by their corresponding Katz centralities according to Figure 3.12.

Chapter 4

Random Graphs

Graphs are frequently used to model real world networks and one purpose of network science is to mimic these structures with very high accuracy. Usually real world networks are not regular. There are no obvious patterns which can be followed to reproduce their topology and the connections between their vertices appear to be completely random. Random networks are studied within network science and they are subjects of the theory of random networks which analyzes methods for building and characterizing them. Networks are seemingly simple structures composed of vertices and links but the way edges are placed in order to connect vertices poses difficulties when reproducing models of complex systems. Identifying clear patterns of similarity among networks is very important because they can be used to establish relationships, hence known rules and principles can be applied to analyze analogous systems. One of the first pioneers of the theory of random graphs was Anatol Rapoport but the major contributions were given by Edgar Nelson Gilbert, Pál Erdős, and Alfréd Rényi [41]. In fact the very first papers on this field were published by Erdős-Rényi and Gilbert on the same year, leading to two different but related definitions which relation we state in Theorem 4.0.1:

Definition 4.0.1 (Erdős-Rényi's $G(N, L)$ Model). *A random Erdős-Rényi graph namely $G(N, L)$ is the probability distribution on the set \mathbb{G} consisting of all possible graphs with N vertices and L edges where each graph $G_i \in \mathbb{G}$ has an equal probability of being chosen.*

For a fixed graph G_0 with N vertices and L edges we have $P(G_0) = \binom{N}{L}^{-1}$.

Definition 4.0.2 (Gilbert's $G(N, p)$ Model). *A random Gilbert's graph namely*

$G(N, p)$ is the probability distribution on the set of all graphs with N nodes where each pair of vertices v, u has a probability p of being connected by an edge.

Since a random Gilbert's graph $G(N, p)$ has no characterization for the edges in its declaration we can denote by $e(G(N, p))$ its number of edges.

Theorem 4.0.1. *A graph generated by $G(N, L)$ model is the same graph generated by $G(N, p)$ model given that it has L edges.*

Proof. To prove the theorem we have to demonstrate that a graph $G(V, E)$ of $|V| = N$ vertices and $|E| = L$ edges has the same probability of being chosen in both models. Clearly $P(G(N, L) = G(V, E)) = \left(\binom{N}{2}\right)^{-1}$ if it is an Erdős-Rényi random graph. On the other hand if it is a Gilbert random graph we have:

$$\begin{aligned} P(G(N, p) = G(V, E) \mid e(G(N, p)) = L) &= \frac{P(G(V, E))}{P(e(G(N, p)) = L)} \\ &= \frac{p^L (p-1)^{\binom{N}{2}-L}}{\left(\binom{N}{2}\right) p^L (p-1)^{\binom{N}{2}-L}} = \left(\binom{N}{2}\right)^{-1}. \end{aligned}$$

Since $G(V, E)$ has the same probability of being chosen in both models we conclude that it is the same graph. \square

In the rest of the chapter when exploiting random graphs and their properties we will mostly refer to the Gilbert's model because it offers easiness of computation and the derivation of the characteristics in random graphs is more intuitive.

4.1 Statistical Properties

If we generate random networks using $G(N, p)$ with fixed parameters we will obtain very different results based on the number of edges that the networks will have and their placement among the N vertices. It is consequently important to determine expected characteristics of the networks that we will acquire as a result of the procedure.

The probability that a randomly generated graph will have exactly m edges, which we can denote by p_m for representation convenience, will be a multiplication of the following terms:

1. The probability that we will have m edges in our trials to link $\frac{(N-1)N}{2}$ pairs of vertices with each other - that is, p^m .

2. The probability that the rest of the vertices will not have edges in-between - that is, $(1 - p)^{\frac{(N-1)N}{2} - m}$.
3. The number of times that we can choose m pairs of nodes to have edges in-between out of $\frac{N(N-1)}{2}$ pairs in total - that is, $\binom{\frac{N(N-1)}{2}}{m}$.

Finally considering all the above we have $p_m = \binom{\frac{N(N-1)}{2}}{m} p^m (1 - p)^{\frac{(N-1)N}{2} - m}$, which is a binomial distribution. It is straight-forward to calculate the expected number of edges in a random graph from $G(N, p)$ based on the properties of the distribution:

$$E(m) = \sum_{m=0}^{\frac{N(N-1)}{2}} m p_m = p \frac{N(N-1)}{2}.$$

We can thus calculate the average degree of a node to be $\langle k \rangle = \frac{2E(m)}{N}$.

Seemingly we can prove that the probability distribution for a randomly chosen vertex to have degree k follows the binomial distribution.

Theorem 4.1.1. *The probability distribution for a graph $G \in G(N, p)$ to have degree k is binomial.*

Proof. Let G be a $G(N, p)$ network and v one of its vertices. The probability of v to have degree k will be: $P(\deg(v) = k) = \binom{N-1}{k} p^k (1 - p)^{N-1-k}$. \square

In the case when $\langle k \rangle \ll N$ the degree distribution can be approximated by a Poisson distribution that is $p_k = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}$. Since most of the real networks satisfy the property $\langle k \rangle \ll N$, Poisson distribution is usually assumed for their degree distributions because it offers computational easiness by making them independent from the size of networks. Even though random networks can be approximated by a Poisson distribution, it is important to notice that most of the real networks do not follow the same trend. For illustrative purposes let's consider the following example where the maximal and minimal degrees of nodes in a random network are calculated as described in [41].

Example 4.1.1. *Consider a random network consisting of $N \approx 7 \times 10^9$ vertices that represent the world's social network (i.e., random society). Let us now find the minimal and maximal degrees that nodes can have assuming that on average every person knows 1000 other people (i.e., $\langle k \rangle = 1000$). As explained in [41] based on the properties of random graphs it is possible to calculate that in a random society the maximal and minimal degrees that a node is expected to have are $k_{\max} = 1185$*

and $k_{min} = 816$ respectively. The dispersion will be $\sigma_k = \sqrt{\langle k \rangle} = 31.62$, which means that the number of connections that most of the people have should be in the range of $\langle k \rangle \pm \sigma_k$ - that is, between 968 and 1032 connections.

Example 4.1.1 shows an important property of large random graphs according to which the degree of most of the nodes is close to $\langle k \rangle$ (i.e., there are few outliers). This is something that does not correspond to the evidences as there is a considerable number of people having more than 1185 connections [41]. The prevalence of the former observation is supported by numerous evidences, for example in Facebook there are many users having more than 5000 friends which is the maximal number of connections that a user is allowed to have. The connections between computers in the Internet, the network of collaborations in science and the protein interaction networks are other examples that suggest the presence of a considerable percentage of high degree nodes, contrary to the expectations predicted by Poisson distribution. Figure 4.1 compares the Poisson distribution curve to the actual degree distributions of the previously mentioned real networks, where the imprecision of Poisson distribution to predict hubs and very low degree nodes can be clearly seen.

Another important characteristic of networks which reveals information on the connectivity of the neighborhood around each node is the *local clustering coefficient*.

Definition 4.1.1 (Clustering Coefficient). *Let $G(V, E)$ be a graph and let $v \in V$, the local clustering coefficient of v is denoted by C_v and defined as the ratio between the number of triangles connected in v with the total number of triples centered around the vertex v .*

The local clustering coefficient of a node measures the tendency of its neighbors to be a clique with each other. In the case of random graphs we calculate it as $C_i = \frac{2\langle L_i \rangle}{k_i(k_i-1)}$, where k_i is the degree of the node and $\langle L_i \rangle = p \frac{k_i(k_i-1)}{2}$ is the expected number of edges between its k_i neighbor nodes. After a few transformations we can express the local clustering coefficient in the form $C_i = p = \frac{\langle k \rangle}{N-1} \approx \frac{\langle k \rangle}{N}$, and from that we reach the following conclusions:

1. As N increases the local clustering coefficients and the network's average clustering coefficient decrease if we keep $\langle k \rangle$ constant.
2. C_i does not directly dependent on the degree of the node.

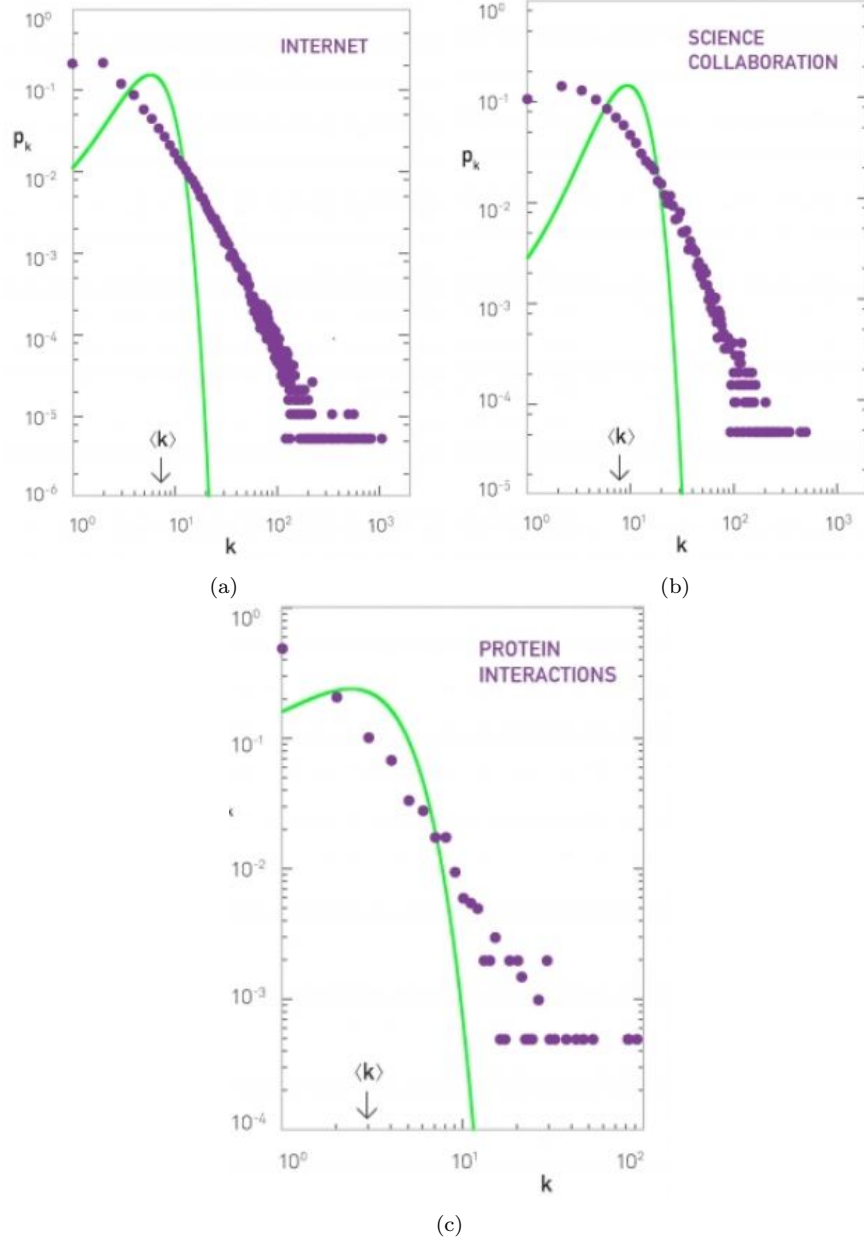


Figure 4.1: The Poisson distribution (green line) and the actual degree distribution of nodes in 3 real networks: (a) Internet, (b) science collaborations, and (c) protein interactions. Taken from [41].

Figure 4.2 shows the discrepancy between the local clustering coefficients of two real networks compared to their respective random networks of the same sizes.

Random networks are interesting mathematical entities to be studied and they reveal useful properties that emerge from random interactions between objects. However, they fail to build accurate models for many real networks that we are

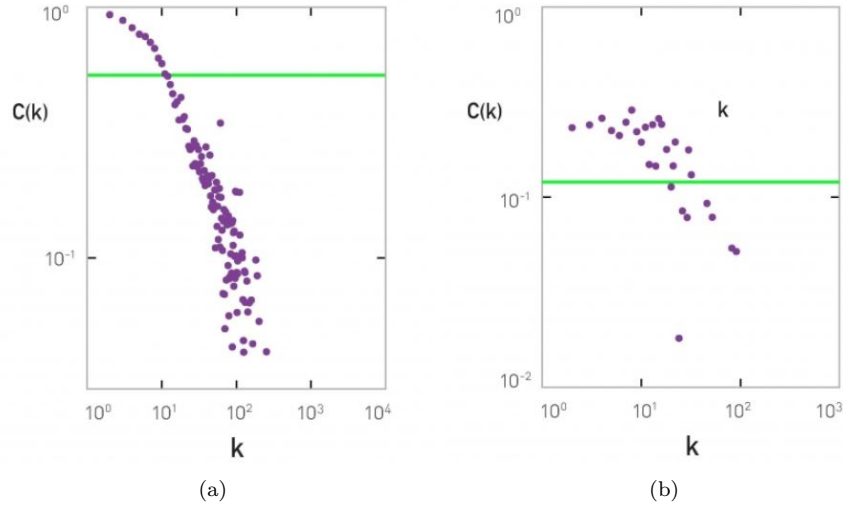


Figure 4.2: Dependency between the local clustering coefficient and the degree of a node on a random network (green line) and on a real network (colored dots) . (a) Network of science collaborations.(b) Network of protein interactions. Taken from [41].

interested to analyze. For this reason additional restrictions are added for the purpose of mimicking the behavior of the real networks. The rest of this chapter will cover the description of two types of network models which unlike Erdős-Rényi's graphs are not completely random, and they share more similarities with networks that are encountered in the real world.

4.2 Small-World

A specific subcategory of random graphs satisfy the property that allows every node to reach any other one by traversing a relatively small number of connecting nodes in-between (i.e., *small-world property*). These types of graphs are called *small-world networks* (i.e satisfying the small-world property) and the property is formally defined as follows:

Definition 4.2.1 (Small-World Property). *A network is said to satisfy the small-world property if the average length ℓ' of the shortest paths between any pair of nodes is smaller then the logarithm of the total number of vertices N :*

$$\lim_{N \rightarrow \infty} \ell' = O(\log N).$$

The networks satisfying the previous definition manifest an interesting at-

tribute which in social networks is referred to as the *small-world phenomenon*. This effect indicates that in a network representing familiarity relations among people (i.e., nodes are connected only if those people know each other), any two strangers are linked with a distance which is orders of magnitude less than the size of the network.

Definition 4.2.2 (Small-World Phenomenon). *The small-world phenomenon or 6 degrees of separation states that any two people on Earth are connected with each other on average by no more than 6 other people who know each other.*

It is not surprising that people living in small geographic areas (e.g., city) can know each other via just a few number of other people in-between, but according to the small-world phenomenon this fact is generally true on average even for people taken from different sides of the planet. There are many evidences which support the validity of the small-world phenomenon. One of them is the famous Milgram's small-world experiment as described in [42]. Analyses that are done to social media networks like Facebook and Twitter provide additional evidences on the existence of this phenomenon.

If we denote by $\langle k \rangle$ the average degree of the network and by $N(d = \ell)$ the number of nodes expected to be at distance ℓ from a random node v then the following relations hold:

- $N_v(d = 1) \approx \langle k \rangle$.
- $N_v(d = 2) \approx \langle k \rangle^2$.
- ...
- $N_v(d = \ell) \approx \langle k \rangle^\ell$.

Theorem 4.2.1. *The number of nodes at distance ℓ from a node v in a random graph is $N_v(d = \ell) \approx \langle k \rangle^\ell$, where $\langle k \rangle$ is the average degree of the nodes in the graph.*

Proof. In a random graph with N vertices for any node we expect to have $p(N - 1)$ nodes connected to it since each of the other $N - 1$ node has probability p of being connected with the initial node, thus:

$$N_v(d = 1) = \sum_{j=1}^{N-1} P(d(v, j) = 1) = p(N - 1) = \langle k \rangle.$$

The expected number of vertices found at distance 2 is:

$$\begin{aligned} N_v(d=2) &= \sum_{j=1}^{N-1} P(d(v,j)=2) = \sum_{l=1}^{N-2} P(d(v,l)=1, d(l,j)=1) \\ &\approx \sum_{l=1}^{N-1} p(N-1) = p^2(N-1)^2 = \langle k \rangle^2. \end{aligned}$$

and in general the expected number of vertices found at distance ℓ will be:

$$\begin{aligned} N_v(d=\ell) &= \sum_{j=1}^{N-1} P(d(v,j)=\ell) = \sum_{l=1}^{N-2} P(d(v,l)=\ell-1, d(l,j)=1) \\ &\approx \sum_{l=1}^{N-1} p^\ell(N-1)^{\ell-1} = p^\ell(N-1)^\ell = \langle k \rangle^\ell. \end{aligned}$$

□

Since we use random graphs as possible models to mimic real networks, it is relevant to know whether the small-world property is true for them. The following theorem gives us the required information.

Theorem 4.2.2. *Any random graph has the small-world property.*

Proof. The expected number of vertices having distances up to ℓ from v will be:

$$C_{D_\ell} \approx \langle k \rangle + \langle k \rangle^2 + \dots + \langle k \rangle^\ell = \frac{\langle k \rangle^{\ell+1} - 1}{\langle k \rangle - 1}.$$

For the sake of simplicity we can approximate $\frac{\langle k \rangle^{\ell+1}}{\langle k \rangle - 1} \approx \langle k \rangle^\ell$. Clearly C_{D_ℓ} cannot take any value and it is bounded by the number of vertices in the graph that is N . As such we can calculate the value of the maximal distance ℓ_{max} between two nodes to be the solution of the equation : $\langle k \rangle^\ell \approx N$ and find that:

$$\ell_{max} \approx \frac{\ln N}{\ln \langle k \rangle}, \quad (4.1)$$

which means that the network has the small-world property according to the Definition 4.2.1. □

The result in (4.1) allows for the following interpretations:

1. Even though (4.1) offers a value for the diameter of the network, it actually corresponds better to the average distance ℓ' between nodes in many real networks as we can see from the Table 4.1.
2. The term $\frac{1}{\ln\langle k \rangle}$ indicates that the average distance is proportionally inverse to the density of the network, namely that if the network is denser (i.e., it has high $\langle k \rangle$) the average distance becomes shorter.

We can now continue the Example 4.1.1 which considers a random network of the size approximately equal to the population of the world and by using (4.1) conclude that in this case the average distance ℓ' between nodes is:

$$\ell' \approx \frac{\ln(7 \times 10^9)}{\ln(10^3)} \approx 4. \quad (4.2)$$

According to Barabási [41], the result obtained in (4.2) is a better estimation to the real average distance between people on Earth compared to Milgram's experiment.

Network	# nodes	# edges	$\langle k \rangle$	ℓ'	ℓ_{max}	$\frac{\ln N}{\langle k \rangle}$
Internet	192244	609066	6.34	6.98	26	6.58
WWW	325729	11497134	4.60	11.27	93	8.31
Power Grid	4941	6594	2.67	18.99	46	8.66
Mobile-Phone Calls	36595	91826	2.51	11.72	39	11.42
Email	57194	103731	1.81	5.88	18	18.4
Science Collaboration	23133	93437	8.08	5.35	15	4.81
Actor Network	702388	29397908	83.71	3.91	14	3.04
Citation Network	449673	4707958	10.43	11.21	42	5.55
E. Coli Metabolism	1039	5802	5.58	2.98	8	4.04
Protein Interactions	2018	2930	2.90	5.61	14	7.14

Table 4.1: Statistical measurements done in different kind of real networks. Data taken from [41].

4.2.1 Watts-Strogatz Model

The Table 4.1 suggests that the properties of random graphs can be useful when analyzing real networks. Figure 4.2 shows a discrepancy between the clustering coefficient of real networks and their analogous random graphs of the same size. Based on these observations Duncan Watts and Steven Strogatz proposed a new

model for building random networks by setting more restrictions on them in order to construct accurate approximations of networks as they are manifested in the real world. Watts and Strogatz built their extension to the *random network model* which is often referred to as the *small-world model*. They based it on the following two arguments:

1. Real networks have the small-world property since the average distance between vertices increases on a logarithmic scale with their sizes.
2. The clustering coefficients of real vs random networks differ in that it is higher in the case of the first for comparable numbers of edges and vertices.

The small-world model uses properties of regular lattices (e.g., rectangular grid graphs) which have high clustering coefficients and high average distance between nodes, and properties of random graphs which have low clustering coefficients and low average distance of nodes. Below we describe the procedure for building networks according to *Watts-Strogatz model*:

1. We start with a set of nodes v_1, v_2, \dots, v_k connected in a circular fashion and we connect each node v_i to all the nodes in its initial n th order zone $Z_n(v)$ (see Figure 4.3 (a)).
2. We then choose with probability p to reconnect each node from the previous configuration to a random node from the set of nodes. At the end of the procedure the initial graph will have changed (see Figure 4.3 (b)).

We notice that the small-world model is dependent from the probability p . In the case when $p = 0$ the graph produced will be a lattice and in the case when $p = 1$ it will be a random graph. All the graphs with desired properties similar to real networks are produced for values of p laying in the open interval between 0 and 1. Another noticeable thing is that the degree distribution is Poisson and as a result hubs like those present in real networks are missing in the predictions.

4.3 Scale-Free

The World Wide Web is an important network that plays a significant role in our daily lives. It is composed of documents which contains URLs that are references to other documents in a giant graph of approximately 10^{12} different nodes that is larger than the network of human neurons with approximately 10^{11} nodes. One of

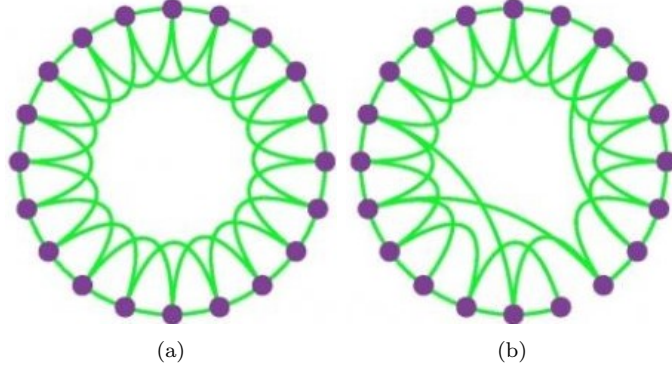


Figure 4.3: The resulting graphs after each step of building a Watts-Strogatz model.(a) Every node inside the ring is directly connected to all the nodes in their second order zone of the initial ring.(b) Every node is rewired with probability p with another one (note that p is low in this case since very few links have changed). Taken from [41].

the first attempts to build a map of WWW was made in 1998 in order to analyze its topology and find whether it shared properties with random graphs. Based on the fact that web pages have links relevant to their content, and given that the content of every page varies from the interest of their designers, it was believed that the structure of the WWW should resemble to that of a randomly generated graph. Figure 4.4 shows the distribution of in and out degrees in the WWW network (normalized by log-log transformations) as compared to the Poisson distribution curve and the power function $k^{-\gamma}$ for some fixed γ . As we can see the degree distribution line of this network does not fit to the Poisson curve, therefore it cannot be a random graph. On the other hand we can observe that the power function $k^{-\gamma}$ for some well chosen γ offers a very accurate approximation to the actual degree distribution of WWW. Networks whose nodes' degree distribution can be approximated by a power function are called *scale-free network*.

Definition 4.3.1. *A network is called scale-free (i.e., has the scale free property) if the number of edges connected to each node follows the power law distribution $P(k) \sim k^{-\gamma}$ for some fixed γ .*

The underlying principle of scale-free networks is that every new node which is added to the network obeys to the rule of *preferential attachment*, namely for every existing node v_i in the network there exists a probability proportional to the number of degrees k_i for sharing an edge with it. More formally if we have a graph $G(V, E)$ the probabilities of a new node v to share an edge with the an existing node v_i will be: $P_i \sim \frac{k_i}{\sum_{j=1}^{|V|} k_j}$.

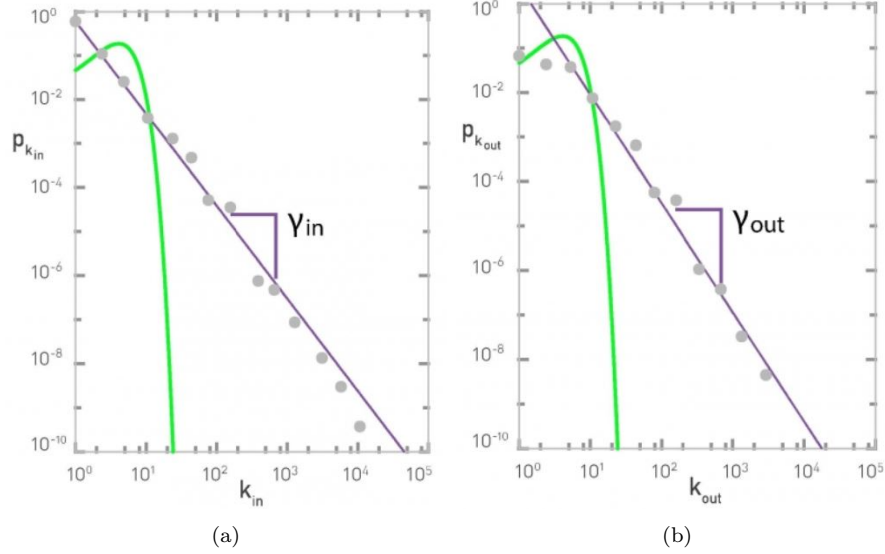


Figure 4.4: The log-log transformation of the degree distribution of nodes in WWW represented by dots, the power law fit function $k^{-\gamma}$ resembling a straight line, and the Poisson distribution curve in green.(a) Represents the in-degree distribution of nodes and $\gamma = \gamma_{in} = 2.1$.(b) Represents the out-degree distribution of nodes and $\gamma = \gamma_{out} = 2.45$. Taken from [41].

It is possible to derive interesting properties of the scale-free networks considering their power law degree distribution. If we have a connected scale-free network and we denote by p_k the probability that a node has exactly k edges then we can write:

$$p_k = Ck^{-\gamma}, \quad (4.3)$$

where C is a normalizing constant given that:

$$\sum_{k=1}^{\infty} p_k = 1. \quad (4.4)$$

Combining (4.3) and (4.4) we obtain:

$$C \sum_{k=1}^{\infty} k^{-\gamma} = 1 \iff C = \frac{1}{\sum_{k=1}^{\infty} k^{-\gamma}} = \frac{1}{\zeta(\gamma)}, \quad (4.5)$$

and $\zeta(\gamma)$ is the *Rieman-zeta function* [43]. Finally the probability for a node to have degree k can be written as $p_k = k^{-\gamma} \zeta(\gamma)^{-1}$.

We can extend the concept of the degree of a node to any positive value for computational convenience, thus the discrete sum in (4.4) will change into (4.6)

and the constant C can be calculated as in (4.7):

$$\int_{k_{min}}^{\infty} p(k)dk = 1, \quad (4.6)$$

$$C = \frac{1}{\int_{k_{min}}^{\infty} k^{-\gamma} dk} = (\gamma - 1)k_{min}^{\gamma-1}, \quad (4.7)$$

and finally:

$$p(k) = (\gamma - 1)k_{min}^{\gamma-1}k^{-\gamma}. \quad (4.8)$$

It is important to notice that the assumption for the degrees of the nodes to have discrete values gives a meaningful interpretation to p_k - that is, the probability of a node to have degree k . On the other hand when considering k to be a positive real number we are only allowed to calculate the probability of a node to have a degree in a continuous interval between two values. For example $\int_{k_1}^{k_2} p(k)dk$ calculates the probability for the node to have a degree in the interval between k_1 and k_2 .

Scale-free networks have been the subject of study in different fields such as biology, social networks, physics and engineering [44, 45, 46]. One of the first observations of scale-free networks was done by Price when he studied the network of scientific papers' citations. In that study he concluded that the degrees of nodes follow a power law distribution and after many attempts he calculated γ to be 3.04. Scale-free property is identified in many other real world networks such as the Internet, power grid in the USA, protein interactions, network of movie actors, collaborations in science, and the network of human sexual contacts.[29]

4.3.1 Barabási-Albert Model

The importance of scale-free networks triggered the need for models which could generate them. In 1999 Barabási and Albert [47] proposed a method for generating scale-free networks called the *Barabási-Albert model*. It is based on two fundamental principles which are assumed to be shared among many real world networks:

1. Networks are destined to grow, namely new nodes are continuously added and connections are created between the new nodes and the existing ones.

2. The connections between the new nodes with the existing ones is not random but obey to the rule of preferential attachment which is a probabilistic mechanism that makes possible the creation of hubs in the network.

The Barabási-Albert model starts with an initial arbitrary connected graph $G_0(V_0, E_0)$ and iterates by continuously adding new nodes. Every node that is added in the network creates links with m existing nodes, such that each node v_i has probability $p(k_i) = k_i(\sum_{j=1}^{|V|} k_j)^{-1}$ of being connected with the new node. According to these simple rules, after adding n new nodes to the initial network we will have $|V| = |V_0| + n$ nodes and $|E| = |E_0| + nm$ edges in the network. Figure 4.5 illustrates the process of building a scale free network using Barabási-Albert model. It starts with an initial network $G_0(V_0, E_0)$ consisting of two nodes connected by an edge, and continuous by adding a new node and associating two new links to it according to the preferential attachment rule during each step. As Figure 4.5 shows it is possible to see the creation of hubs by merely observing the first few iterations of the process.

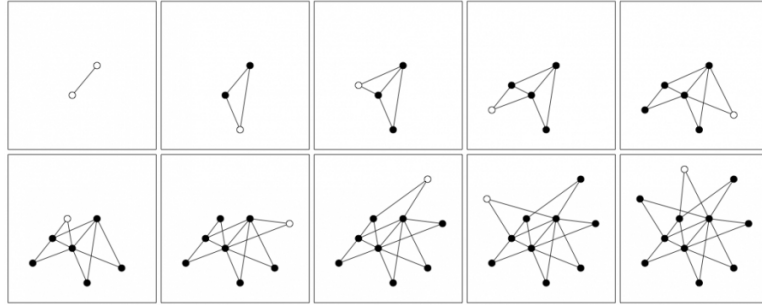


Figure 4.5: The evolution of a Barabási-Albert model during the consecutive addition of 9 new nodes. In this case $G_0(V_0, E_0)$ consists of two nodes connected by an edge. During each step we add a new node denoted by an empty circle, and create two links for it. Taken from [41].

4.3.2 Barabási-Ravasz-Vicsek Model

Stochastic methods for generating scale-free graphs like Barabási-Albert are intuitive in that they mimic very well the features observed in the real world networks. Even so, it is difficult to obtain a visual understanding of scale-free networks based on their constructiveness as they are based on randomness. In order to attain a more concrete insight of scale-free networks and the relations between vertices, Barabási, Ravasz, and Vicsek created a deterministic model to generate scale-free

networks [48]. This model constructs the network iteratively as it is described below:

Step 0: We start the network with one single vertex which we denote as the root of the network.

Step 1: We complement the initial single node network with two additional nodes and connect them to the root.

Step 2: We add two other networks identical to the one which is described in the previous step and we connect the bottom vertices of the newly added networks to the root.

We can thus generalize the n th step as follows:

Step n : We add two other networks identical to the one which is described in the step $n - 1$ consisting of 3^{n-1} nodes and connect the 2^n bottom vertices of the newly added networks to the root.

Figure 4.6 offers a visual explanation of the process described above. We conclude this section by proving the following theorem of interest.

Theorem 4.3.1. *The nodes' degree distribution of Barabási-Ravasz-Vicsek model follows a power law distribution.*

Proof. In order to prove that the model is scale-free we should demonstrate that the degree distribution of the nodes in the network follows the power law distribution. It is clear that the tail of the distribution will be determined by the number of hubs, as such it is enough to focus on the most connected nodes in order to prove the scale-free property of the model. Let's suppose we are on step i of the construction of the network according to the model definition, and let's focus on the hub which is the root node that will have $2^{i+1} - 2$ links. On the next step of the development (i.e., step $i + 1$) we will have $\frac{2}{3} \cdot 3^{(i+1)-i} = 2$ hubs with $2^{i+1} - 2$ links. In the n th step the number of nodes having the same degree as the root node of step i will be $\frac{2}{3} \cdot 3^{n-i}$. If we denote by $P(k)$ the number of nodes having degree k in the network at step n (i.e., degree distribution function) we will have:

$$P(2^{i+1} - 2) = \frac{2}{3} \cdot 3^{n-i}. \quad (4.9)$$

Let's suppose $k = 2^{i+1} - 2$ and solve it for i , thus yields $i = \frac{\ln(\frac{k}{2}+1)}{\ln 2}$. Substituting $2^{i+1} - 2$ with k in the distribution function in (4.9) will yield:

$$P(k) = \frac{2}{3} \cdot 3^n \cdot 3^{-\frac{\ln(\frac{k}{2}+1)}{\ln 2}} \sim 3^{-\frac{\ln(\frac{k}{2}+1)}{\ln 2}} \sim \left(\frac{k}{2} + 1\right)^{-\frac{\ln 3}{\ln 2}} \sim k^{-\frac{\ln 3}{\ln 2}}. \quad (4.10)$$

From (4.10) it follows that the network is scale-free and $\gamma = \frac{\ln 3}{\ln 2}$. □

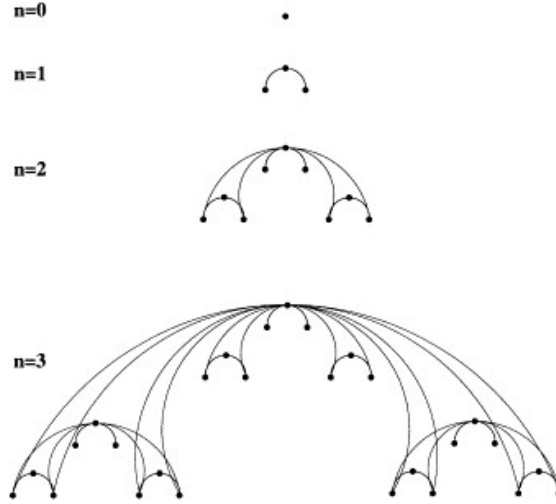


Figure 4.6: The evolution of Barabási-Ravasz-Vicsek model. Taken from [48].

4.4 Random Graphs Generation

In this section we generate 3 networks of 50 nodes according to each of the random models that we previously discussed, namely Erdős-Rényi, Watts-Strogatz, and Barabási-Albert. We analyze each of them and examine the properties that they exhibit as compared to what is already mentioned in this chapter. The generation of the models is done using Networkx [49]. The visual representation of the networks is done with Cytoscape [15], and it can be seen in Figure 4.7. Table 4.2 represents the main metrics of the topology for the generated networks computed with NetworkAnalyzer which is a Cytoscape [15] module that computes a comprehensive set of structural metrics for undirected and directed networks [50]. The rest of this section is dedicated to the description of the networks produced according to each of the models. We also compare some of the results in Table 4.2 measured by NetworkAnalyzer with the actual theoretical expectations.

Erdős-Rényi graph - The network was generated by setting a probability

$p = 0.08$ for the existence of an edge between any two of the 50 vertices of the network. As a result the network produced has 98 edges which is exactly the 8% of the maximal number of edges that a network of 50 vertices can have - that is, $\binom{50}{2} = 1225$. The average shortest path's length is $2.962 < \ln(50) \approx 3.91$ which gives the small-world property to the network. The average degree of the nodes is $\langle k \rangle = (|V| - 1)p = 0.08(50 - 1) = 3.98$. Figure 4.8 (a) shows the histogram of the degree distribution for the network, which is roughly Poisson. There is a clear absence of hubs as predicted by the theory.

Watts-Strogatz graph - The network was generated by starting with a ring topology of 50 vertices, where each vertex is connected to its 6th nearest neighbors. Every vertex of the initial graph was set a probability $p = 0.6$ of being rewired to a randomly selected vertex. The process of rewiring does not change the number of edges which remains 150 or the average degree $\langle k \rangle = 6$ that are both based on the set up of the initial ring configuration. The average shortest path's length is $2.962 < \ln(50) \approx 3.91$ which indicates the small-world property. The degree distribution is roughly Poisson as it can be seen from Figure 4.8 (b) and as a consequence hubs are missing from the network.

Barabási-Albert graph - The network was generated by starting with 3 nodes and introducing 47 other vertices consecutively, in each step every new vertex would connect with 3 existing vertices according to the rule of preferential attachment, thus resulting in a total of 141 edges with an average degree $\langle k \rangle = 5.64$. The network has the small-world property because the shortest path is $2.64 < \ln(50) \approx 3.91$. The degree distribution follows the power law as shown in Figure 4.8 (b) and the presence of hubs is evident by observing the gap which exists between the highest degree frequency bar with all the others.

	Erdős-Rényi	Watts-Strogatz	Barabási-Albert
Number of nodes	50	50	50
Number of edges	98	150	141
Clustering coefficient	0.109	0.098	0.276
Network diameter	6	4	4
Degree centrality index	0.087	0.085	0.369
Average shortest path	2.926	2.296	2.261
Average degree of nodes	3.92	6	5.64
Network density	0.08	0.122	0.115

Table 4.2: Analysis of the topology for each generated network.

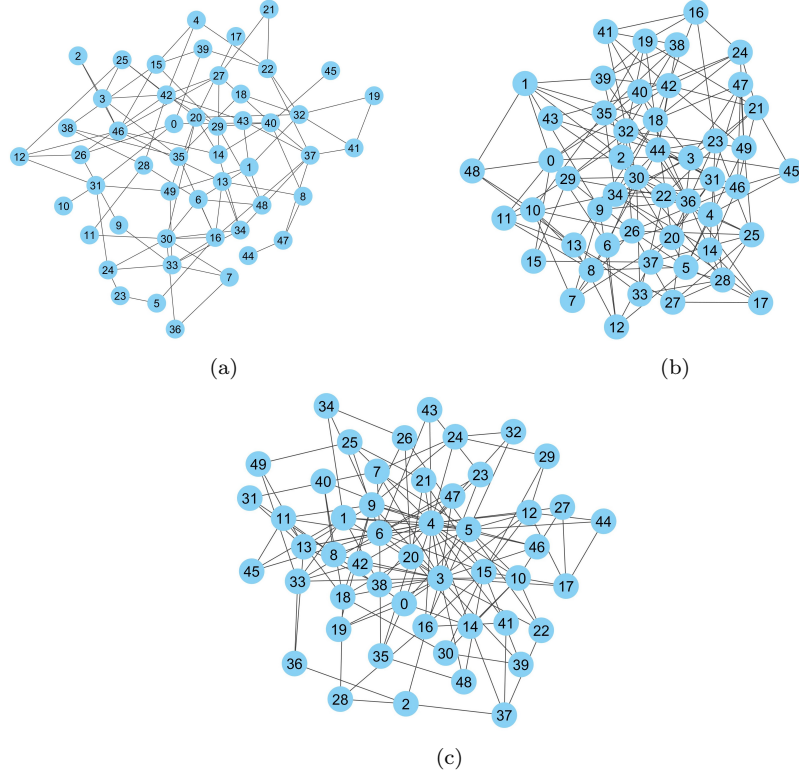


Figure 4.7: Random network models generated with Networkx [49] (a) Erdős-Rényi, (b) Watts-Strogatz and (c) Barabási-Albert models.

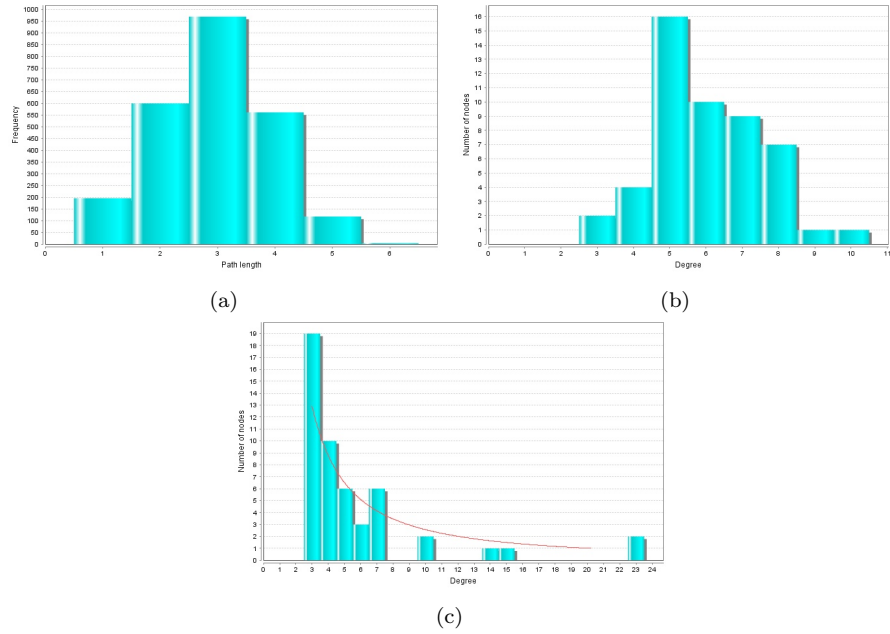


Figure 4.8: The the degree distributions of nodes in the networks generated according (a) Erdős-Rényi, (b) Watts-Strogatz and (c) Barabási-Albert models.

Chapter 5

Methodologies Used

In order to test the effectiveness of centrality methods in the identification of essential genes in multiple myeloma (MM) networks we built 3 cancer networks based on the dataset provided by [51]. Additionally we applied these methods to the drug target nodes, namely the input genes, in order to see whether they display interesting centrality patterns that our methods can predict. In this chapter we will discuss the data and the procedure that we followed in order to build the networks and analyze them. We compared the performance of every method in each network based on the number of essential and drug target genes that are found in the list of the top 100 highest scoring genes according to each centrality method.

5.1 Description and Analysis of Data

We made use of the data which contain information of the mutated genes that were observed from the genetic sequence analysis of 203 patients diagnosed with MM. The data which was gathered and used by Lohr et al. [51] consists of detailed information regarding mutated genes that were found in the tumor samples of the patients, as well as information about general characteristics of the patients. Table 5.1 contains a brief statistical summary of the data present in this dataset.

Number of tumor samples	203
Number of mutations	11017
Number of genes mutated	14562

Table 5.1: Summary of the data in [51].

We then identified 70 genes which are essential for the survival of the MM cells - that is, removing those genes would cause the cell to die. Table 5.3 represents the MM essential genes for survival according to [52].

AGTRAP	CUL9	IKZF3	NDC80	PSMA4	RPL38	TRIM68
NFKB1	PSMA6	RRM1	TUBGCP6	AURKB	EFNA2	IRF4
UBB	EIF3C	KIF11	NFKB2	PSMC3	RSF1	CARS
PSMC4	SF3A1	UBQLNL	CCND2	EIF4A3	KIF18A	NUF2
KIFC2	PCDH18	PSMC5	SLC25A23	ULK3	CDK11	GNRH2
SNRPA1	USP36	CDK11A	GPR77	LEPROT	PIM2	RAB11A
PLK1	RELA	SNW1	USP8	CKAP5	CDK11B	HIP1
WBSCR22	COPB2	IK	MCL1	PRPF8	RELB	TNK2
RGAG1	TPMT	WEE1	IKBKB	MED14	PSMA1	MAF
MED15	PSMA3	RPL27	TRIM21	XPO1	CSNK1A1	IKZF1

Table 5.2: MM essential genes for survival. Taken from [52]

There are in total 27 MM genes to the best of our knowledge which can be targeted by existing MM drugs. Unfortunately only two (i.e., NFKB1 and XPO1) out of the 70 MM essential genes for survival are known to be target of MM standard drugs used for putative therapies. Below we list the MM target genes as described by [52]:

ANXA1	CRBN	HSD11B1	NOS2	NR3C1	PSMB2	PSMB9
TNF	TUBA4A	CD38	FGFR2	NFKB1	NR0B1	PSMB1
PSMB5	PTGS2	TNFSF11	TUBB	CDH5	GSR	NOLC1
NR1I2	PSMB10	PSMB8	SLAMF7	TOP2A	XPO1	

Table 5.3: MM drug target genes.

Sanchez and Petre [52] built 3 PPI networks consisting of the mutated genes corresponding to the 3 different MM samples found in [51], namely MM-0191, MM-0343, and MM-0389 according to a procedure as it is described in [52]. For the purpose of our analysis we will use a slightly improved version of the network for MM-0191, and two other networks corresponding to MM-0343 and MM-0389 that all consist of only one connected component per network. Table 5.5 contains a statistical description for each of the networks that we consider in this chapter.

	MM-0028	MM-0038	MM-0191
Number of edges	7473	9305	11280
Number of nodes	1541	1876	2226
Number of mutated genes	36	117	218
Number of essential MM genes	65	65	65
Number of drug target MM genes	27	27	27
Network diameter	10	10	8
Average shortest path	3.684	3.663	3.565
Average degree of nodes	7.563	7.319	7.045

Table 5.4: Statistical analysis of MM generated networks.

5.2 Centrality Analysis of the Networks

We performed a centrality analysis for each of the PPI networks and ranked the nodes according to their degree centrality, closeness centrality, betweenness centrality, harmonic centrality, degree prestige, eigenvector-based prestige, and Katz prestige accordingly. The centrality analysis of the networks was done using Networkx [49] which can read the graphml files of the networks that we generated with Cytoscape [15] and initialize network objects from them. The degree distribution frequency analysis was done with NetworkAnalyzer. Table 5.5 shows the results of our analysis in each network. There is a clear tendency for degree centrality, closeness centrality, and betweenness centrality to assign higher scores to the essential genes and input genes. On average approximately 70% of the 65 essential genes and 74% of the 27 input genes present in the networks belong to the set of the top 100 highest ranked genes according to these centralities. On the other hand harmonic centrality, eigenvector-based prestige and Katz prestige performed weaker in predicting important genes - that is, on average approximately 35% of the essential genes and 49% of the input genes belong to the set of their top 100 highest ranked genes. In either case all centrality methods except for Katz prestige displayed higher accuracy in predicting input genes compared to essential genes. In all the networks the top ranked genes by degree centrality and degree prestige correspond to an essential gene. There is a clear indication that most of the essential genes and drug target genes have more central positions in the topology of the network. Below we give a detailed description on the centrality analysis results for each of the networks.

MM-0028			
	TRG	NEG	NIG
Degree centrality	RELA	48 (73.8%)	22 (81.5%)
Closeness centrality	UBC	47 (72.3%)	19 (70.3%)
Betweenness centrality	UBC	47 (72.3%)	23 (85.2%)
Harmonic centrality	UBC	36 (55.4%)	18 (66.7%)
Degree prestige	RELA	46 (70.7%)	21 (77.7%)
Eigenvector-based prestige	PSMC5	21 (32.3%)	17 (63%)
Katz prestige	CUL9	16 (24.6%)	0 (0%)
MM-0038			
	TRG	NEG	NIG
Degree centrality	RELA	45 (69.2%)	21 (77.8%)
Closeness centrality	UBC	45 (69.2%)	19 (70.4%)
Betweenness centrality	UBC	44 (67.7%)	19 (70.4%)
Harmonic centrality	UBC	35 (53.8%)	18 (66.7%)
Degree prestige	RELA	45 (69.2%)	21 (77.8%)
Eigenvector-based prestige	PSMC5	20 (30.7%)	17 (63%)
Katz prestige	DACH1	10 (15.4%)	6 (22.2%)
MM-0191			
	TRG	NEG	NIG
Degree centrality	SNW1	42 (64.6%)	20 (74.1%)
Closeness centrality	UBC	45 (69.2%)	20 (74.1%)
Betweenness centrality	UBC	40 (61.5%)	17 (63%)
Harmonic centrality	UBC	38 (58.5%)	19 (70.4%)
Degree prestige	RELA	43 (66.2%)	20 (74.1%)
Eigenvector-based prestige	UBC	20 (30.8%)	16 (60%)
Katz prestige	UBE2N	8 (12.3%)	6 (22.2%)

Table 5.5: Centrality analysis of our 3 MM networks. TRG is the top ranked gene. NEG and NIG are the numbers of essential genes and input genes found in the list of the top 100 ranked genes accordingly.

5.2.1 MM-0028 Network

Tumor sample MM-0028 has only 36 different mutated genes and it is the smallest network. Its corresponding network has 1541 nodes. The ranking of the nodes according to degree centrality contains the highest number of essential genes in the set of top 100 highest scoring genes - that is, 48 out of 65 or 73.8%. The ranking of the nodes according to betweenness centrality contains the highest number of drug target genes in the set of top 100 highest scoring genes - that is 23 out of 27 or 85.2%. The average length of shortest paths is 3.684. Considering

the low average distance between vertices compared to the number of vertices, $\ln(1541) \approx 7.34 > 3.684$ the network is small-world. Figure 5.1 shows the out-degree distribution of the network and a power law function with coefficient of determination $R^2 = 0.802$. The correlation between the degree distribution and the corresponding points on the power law function is 0.989. According to this analysis the network is scale-free.

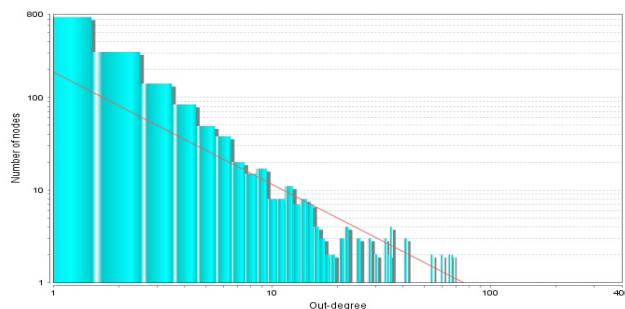


Figure 5.1: The histogram of out-degree distribution for MM-0028 network after applying log-log transform and the power law function $y = 186.25x^{-1.208}$.

5.2.2 MM-0038 Network

Tumor sample MM-0038 has 117 different mutated genes. Its corresponding network has 1876 nodes. Degree centrality, closeness centrality and degree prestige could rank 45 out of 65 essential genes or 69.2% in their top 100 highest scoring genes. On the other hand the maximum number of drug target genes - that is, 19 out of 27 or 77.8% can be found only in the list of top 100 highest scoring genes ranked according to degree centrality and degree prestige. The average of shortest paths is 3.663 and $\ln(1876) \approx 7.536 > 3.663$, therefore it is a small-world network. Figure 5.2 shows the out-degree distribution of the network and a power law function with coefficient of determination $R^2 = 0.811$. The correlation coefficient between the degree distribution and the power law function is 0.993. Given that the degree distribution can be accurately approximated by a power law function it can be said that the network is scale-free.

5.2.3 MM-0191 Network

Tumor sample MM-0191 has 218 different mutated genes and its corresponding network has 2226 nodes. Closeness centrality is the best performing method for

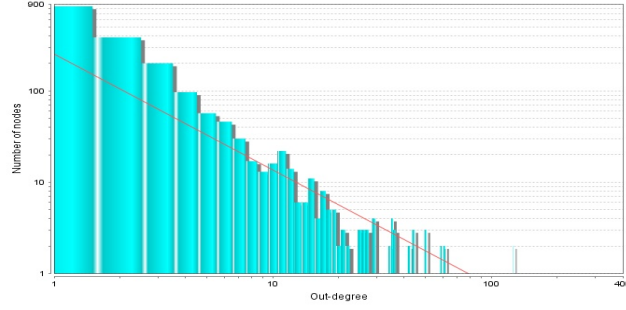


Figure 5.2: The histogram of out-degree distribution for MM-0038 network after applying log-log transform and the power law function $y = 253.16x^{-1.268}$.

finding the highest number of essential genes in its top 100 ranked genes - that is, 45 out of 65 genes or 69.2%. However degree centrality, closeness centrality and degree prestige performed equally in finding drug target genes, by identifying 20 out of 27 or 74.1%. The average distance between nodes is 3.565 which is lower than $\ln(2226) \approx 7.7$, thus letting it to have the small-world property. Figure 5.3 shows the out-degree distribution of the network against a power law function and as in the previous two networks we have very high values of $R^2 = 0.811$ and a correlation coefficient equal to 0.995 which indicate the scale-free property of the network.

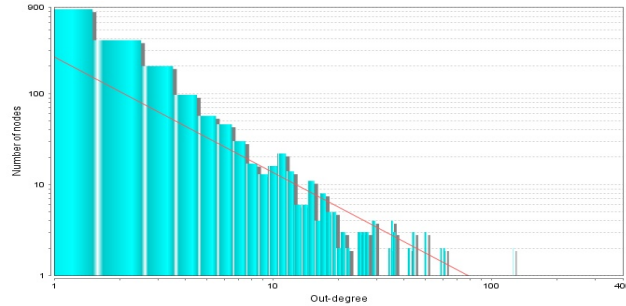


Figure 5.3: The histogram of out-degree distribution for MM-0038 network after applying log-log transform and the power law function $y = 306.34x^{-1.299}$.

5.3 Interpretations and Conclusion

The application of centrality methods in the MM tumor networks yielded interesting properties that essential genes and drug target genes have in terms of topological positions within the networks. According to our analysis there is a clear indication that these genes are characterized by higher centrality scores. We

noticed that degree centralities, path centralities and proximity centralities performed in all cases better than spectral centralities in giving high scores to essential and drug target genes. To our surprise there was a significant difference in the number of essential and drug target genes found in the top high scoring genes of closeness and harmonic centrality, even though they belong to the same centrality group (i.e., proximity centralities) and furthermore they are very similar in their definitions as we saw from Chapter 3. The top ranked genes according to each of the 7 centrality methods in all networks correspond to essential genes with except to 3 genes, namely UBC, DACH1, and UBE2N with an exceptional occurrence of UBC which is listed 10 times as the highest scoring gene by different methods in all networks. Given the unusual high frequency of being selected as a top ranked gene and the fact that in almost all the other cases top ranked genes correspond to one of the genes in the essential genes list (i.e., RELA, PSMC5, CUL9, SNW1), we did research on its implications according to recent medical studies. UBC encodes the protein polyubiquitin-C that is involved in the process of ubiquitylation which is responsible for protein degradation, DNA repair, transcription, protein trafficking, cell-cycle regulation, and signal transduction [53]. Ubiquitylation is a mechanism that is essential for appropriate cell survival and function, during which the attachment of ubiquitin to a target protein happens [54]. Ubiquitin is encoded by 4 genes, two of which, namely UBC and UBB, encode polyproteins and are stress-regulated genes that are crucial for keeping ubiquitin levels stable under stress conditions [55]. Considering the strong connection between UBC and UBB and based on the fact that UBB is a MM essential gene and is ranked among the top highest scoring genes according to the centrality methods, it is reasonable to think that UBC might also be an essential gene given that the list of essential genes for MM is not exhaustive. The inhibition of DACH1 has resulted in repression of tumorigenesis in pancreatic cancer and it decreased the progression of myeloid leukemia [56, 57]. The last top ranked gene UBE2N which was not listed as an essential gene of MM was found to be essential for the death of neuroblastoma cells [58]. Finally we were able to prove that our MM networks are scale-free and furthermore exhibit the small-world property.

Chapter 6

Discussion

Biological systems are highly complex structures that constitute a challenge when it comes to finding an appropriate computational model that could mimic them. The representational difficulty of these systems necessitates the existence of different modeling techniques which consist of and is not limited to ODE models, Boolean models, and network models. In this thesis we considered networks as a possible alternative of modeling relations between genes in MM tumor samples. As opposed to the standard approach where observations about the system are made in the first place and then a model is built to formalize this observations and analyze the system on their basis, our approach is reversed. Namely, we try to find relevance of the pure network's concept of node centrality and explore whether it has implications on aspects of the system that were not directly taken in consideration when building the models.

The importance of network centrality has been extensively studied for a long time in the context of social networks but fewer studies exist to study its involvement in the case of biological networks. We made a thorough review of the main methods to measure degree, path, proximity, and spectral centralities by giving appropriate examples for their usage. We believe that this assessment provides the necessary intuition and details for researchers who are unfamiliar with graph theory and network science to make use of these tools in the study of diseases. Additionally the notion of random networks was discussed, several models for building them were examined and their properties were compared with the MM networks built from 3 tumor samples.

We hypothesized that high centrality of nodes in networks of genes might indicate a particular significance in the role that those genes play in the mutated

cell. We tested our hypothesis by measuring different node centralities in MM gene networks and were able to find a significant correlation between the centrality score of a gene and its potential of being essential to the mutated cell, or a drug target - that is, almost all of the essential and drug target genes could be found in the list of top 100 scored genes according to some centrality measure. The highest ranked gene according to each centrality did in all cases belong to our initial list of essential genes with the exception of UBC, DACH1 and UBE2N. Our investigation on these 3 exceptional genes showed that there are reasons to suspect a possible involvement in the vitality of MM cells and further research should be made to consider their potential presence in the list of MM essential genes. We believe that the correspondence of high centrality with the presence in the list of essential and drug target genes is not a mere coincidence. More studies are needed to investigate the strength of this relation but the implications can be significant in the identification of important genes via mere computation and will help solving a variety of problems such as target controllability at least.

Bibliography

- [1] Irina Gribkovskaia, Øyvind Halskau Sr, and Gilbert Laporte. The bridges of Königsberg—a historical perspective. *Networks: An International Journal*, 49(3):199–203, 2007.
- [2] National Research Council. *Network Science*. The National Academies Press, Washington, DC, 2005.
- [3] Krishna Kanhaiya, Vladimir Rogojin, Keivan Kazemi, Eugen Czeizler, and Ion Petre. Netcontrl4biomed: A pipeline for biomedical data acquisition and analysis of network controllability. *BMC Bioinformatics*, 19(7):3–12, 2018.
- [4] Jiajie Peng, Tao Wang, Jianping Hu, Yadong Wang, and Jin Chen. Constructing Networks of Organelle Functional Modules in Arabidopsis. *Current Genomics*, 17:427–438, 08 2016.
- [5] Sabine Pérès, Liza Felicori, and Franck Molina. Elementary flux modes analysis of functional domain networks allows a better metabolic pathway interpretation. *PLoS One*, 8(10):1–9, 10 2013.
- [6] Fabrizio Vecchio, Francesca Miraglia, and Paolo Maria Rossini. Connectome: Graph theory application in functional brain network architecture. *Clinical Neurophysiology Practice*, 2:206–213, 2017.
- [7] https://assays.cancer.gov/available_assays?wp_id=wp78.
- [8] Linton C. Freeman. Centrality in social networks conceptual clarification. *Social Networks*, 1(3):215–239, 1978.
- [9] Matthew W. Hahn and Andrew D. Kern. Comparative Genomics of Centrality and Essentiality in Three Eukaryotic Protein-Interaction Networks. *Molecular Biology and Evolution*, 22(4):803–806, 12 2004.

- [10] Arzucan Özgür, Thuy Vu, Güneş Erkan, and Dragomir R Radev. Identifying gene-disease associations using centrality on a literature mined gene-interaction network. *Bioinformatics*, 24(13):i277–i285, 2008.
- [11] Jihua Ran, Hui Li, Jianfeng Fu, Ling Liu, Yanchao Xing, Xiumei Li, Hongming Shen, Yan Chen, Xiaofang Jiang, Yan Li, et al. Construction and analysis of the protein-protein interaction network related to essential hypertension. *BMC Systems Biology*, 7(1):32, 2013.
- [12] Maliackal Poulo Joy, Amy Brock, Donald E Ingber, and Sui Huang. High-betweenness proteins in the yeast protein interaction network. *BioMed Research International*, 2005(2):96–103, 2005.
- [13] Sanjoy Dasgupta, Christos H Papadimitriou, and Umesh Virkumar Vazirani. *Algorithms*. 2006.
- [14] <http://ugrad.stat.ubc.ca/r/library/snadata/html/florentine.html>.
- [15] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S Baliga, Jonathan T Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, 2003.
- [16] Peter V. Marsden. Network centrality, measures of. In James D. Wright, editor, *International Encyclopedia of the Social & Behavioral Sciences (Second Edition)*, pages 532–539. Elsevier, Oxford, second edition edition, 2015.
- [17] Leigh Metcalf and William Casey. Chapter 5-graph theory. In Leigh Metcalf and William Casey, editors, *Cybersecurity and Applied Mathematics*, pages 67–94. Syngress, Boston, 2016.
- [18] Deepak Sharma and Avadhesha Surolia. *Degree Centrality*, pages 558–558. Springer New York, 2013.
- [19] Kousik Das, Sovan Samanta, and Madhumangal Pal. Study on centrality measures in social networks: a survey. *Social Network Analysis and Mining*, 8(1):13, 2018.
- [20] James Powell and Matthew Hopkins. 19-graph analytics techniques. In James Powell and Matthew Hopkins, editors, *A Librarian’s Guide to Graphs, Data*

and the Semantic Web, Chandos Information Professional Series, pages 167–174. Chandos Publishing, 2015.

- [21] Dirk Koschützki and Falk Schreiber. Centrality analysis methods for biological networks and their application to gene regulatory networks. *Gene Regulation and Systems Biology*, 2:GRSB.S702, 2008.
- [22] U Kang, Spiros Papadimitriou, Jimeng Sun, and Hanghang Tong. *Centralities in Large Networks: Algorithms and Observations*, pages 119–130.
- [23] Gert Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.
- [24] Ulrik Brandes. A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology*, 25(2):163–177, 2001.
- [25] Linton C Freeman. A set of measures of centrality based on betweenness. *Sociometry*, pages 35–41, 1977.
- [26] R. Sahoo, T.S. Rani, and S.D. Bhavani. Chapter 17 - Differentiating Cancer From Normal Protein-Protein Interactions Through Network Analysis. In Quoc Nam Tran and Hamid R. Arabnia, editors, *Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology*, Emerging Trends in Computer Science and Applied Computing, pages 253–269. Morgan Kaufmann, Boston, 2016.
- [27] Meghana Nasre, Matteo Pontecorvi, and Vijaya Ramachandran. Betweenness Centrality – Incremental and Faster, 2013.
- [28] Yannick Rochat. Closeness Centrality Extended to Unconnected Graphs: the Harmonic Centrality Index. 2009.
- [29] M.E.J. Newman. The structure and function of networks. *Computer Physics Communications*, 147(1):40–45, 2002. Proceedings of the Europhysics Conference on Computational Physics Computational Modeling and Simulation of Complex Systems.
- [30] Paolo Boldi and Sebastiano Vigna. Axioms for centrality. *Internet Mathematics*, 10(3-4):222–262, 2014.

- [31] Raj Kumar Pan and Jari Saramäki. Path lengths, correlations, and centrality in temporal networks. *Physical Review E*, 84(1):016105, 2011.
- [32] Edith Cohen and Haim Kaplan. Spatially-decaying aggregation over a network. *Journal of Computer and System Sciences*, 73(3):265–288, 2007.
- [33] Eugenio Angriman. Efficient computation of harmonic centrality on large networks: theory and practice. 2016.
- [34] Dirk Koschützki. *Network Centralities*, chapter 4, pages 65–84. Wiley Online Library, 2007.
- [35] Yizhou Sun and Jiawei Han. *Ranking Methods for Networks*, pages 1488–1497. Springer New York, New York, NY, 2014.
- [36] Aidong Zhang. *Protein interaction networks: computational analysis*. Cambridge University Press, 2009.
- [37] Ernesto Estrada. Virtual identification of essential proteins within the protein interaction network of yeast. *Proteomics*, 6(1):35–40, 2006.
- [38] Ernesto Estrada and Juan A Rodriguez-Velazquez. Subgraph centrality in complex networks. *Physical Review E*, 71(5):056103, 2005.
- [39] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.
- [40] Kurt Foster, Stephen Muth, John Potterat, and Richard Rothenberg. A Faster Katz Status Score Algorithm. *Computational & Mathematical Organization Theory*, 7:275–285, 12 2001.
- [41] Albert-László Barabási. *Network Science*. Cambridge University Press, 2016.
- [42] Jon Kleinberg. The Small-World Phenomenon: An Algorithmic Perspective. In *In Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 163–170, 2000.
- [43] <https://www.britannica.com/science/riemann-zeta-function>.
- [44] Albert-László Barabási and Zoltán N Oltvai. Network biology: understanding the cell’s functional organization. *Nature Reviews Genetics*, 5(2):101–113, February 2004.

- [45] Reuven Cohen, Shlomo Havlin, and Daniel ben Avraham. *Structural properties of scale-free networks*, chapter 4, pages 85–110. John Wiley & Sons, 2005.
- [46] Pawel Sobkowicz. Modelling Opinion Formation with Physics Tools: Call for Closer Link with Reality. *Journal of Artificial Societies and Social Simulation*, 12(1):11.
- [47] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512, 1999.
- [48] Albert-László Barabási, Erzsébet Ravasz, and Tamás Vicsek. Deterministic scale-free networks. *Physica A: Statistical Mechanics and its Applications*, 299(3):559–564, 2001.
- [49] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using Networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.
- [50] http://manual.cytoscape.org/en/stable/Network_Analyzer.html#analyze-network.
- [51] Carter SL Cruz-Gordillo P Lawrence MS Auclair D Sougnez C Knoechel B Gould J Saksena G Cibulskis K McKenna A Chapman MA Straussman R Levy J Perkins LM Keats JJ Schumacher SE Rosenberg M Getz G Golub TR. Lohr JG, Stojanov P. Widespread genetic heterogeneity in multiple myeloma: Implications for targeted therapy. *Cancer Cell*, 25(1):91–101, 2014.
- [52] Jose Angel Sanchez Martin and Ion Petre. Network controllability analysis of three multiple-myeloma patient genetic mutation datasets. *Fundamenta Informaticae*, 2020, to appear.
- [53] Kaisa Haglund and Ivan Dikic. Ubiquitylation and cell signaling. *The EMBO Journal*, 24(19):3353–3359, 2005.
- [54] Diane L Haakonsen and Michael Rape. Ubiquitin levels: the next target against gynecological cancers? *The Journal of Clinical Investigation*, 127(12):4228–4230, 2017.

- [55] Kwon-Yul Ryu, René Maehr, Catherine A Gilchrist, Michael A Long, Donna M Bouley, Britta Mueller, Hidde L Ploegh, and Ron R Kopito. The mouse polyubiquitin gene *ubc* is essential for fetal liver development, cell-cycle progression and stress tolerance. *The EMBO Journal*, 26(11):2693–2706, 2007.
- [56] Xiao-Na Bu, Chan Qiu, Chuan Wang, and Zheng Jiang. Inhibition of *dach1* activity by short hairpin rna represses cell proliferation and tumor invasion in pancreatic cancer. *Oncology Reports*, 36(2):745–754, 2016.
- [57] Jae-Woong Lee, Hyeng-Soo Kim, Seonggon Kim, Junmo Hwang, Young Hun Kim, Ga Young Lim, Wern-Joo Sohn, Suk-Ran Yoon, Jae-Young Kim, Tae Sung Park, Kwon Moo Park, Zae Young Ryoo, and Sanggyu Lee. *Dach1* regulates cell cycle progression of myeloid cells through the control of cyclin d, cdk 4/6 and p21cip1. *Biochemical and Biophysical Research Communications*, 420(1):91–95, 2012.
- [58] J Cheng, YH Fan, X Xu, Hong Zhang, J Dou, Y Tang, X Zhong, Y Rojas, Y Yu, Y Zhao, et al. A small-molecule inhibitor of *ube2n* induces neuroblastoma cell death via activation of p53 and jnk pathways. *Cell Death & Disease*, 5(2):e1079–e1079, 2014.

Appendix A

Centrality Analysis Results for MM Networks

A.1 MM-0028 Network

Degree Centrality

Top gene: RELA

Number of input genes for degree centrality: 26

Input genes for degree centrality: [NFKB1, NR3C1, TUBB, NOS2, XPO1, TNF, PSMB1, PSMB5, PSMB2, TOP2A, TUBA4A, FGFR2, PSMB8, PSMB9, ANXA1, PTGS2, NR1H2, NOLC1, PSMB10, TNFSF11, CDH5, NR0B1, CRBN, GSR, HSD11B1, CD38]

Number of essential genes for degree centrality: 60

Essential genes for degree centrality: [RELA, SNW1, NFKB1, IKBKB, EIF4A3, PLK1, PSMA3, UBB, PSMC5, AURKB, XPO1, PSMA6, PSMC3, PSMA4, NFKB2, PSMC4, PRPF8, CSNK1A1, SF3A1, RELB, TNK2, MED14, IK, SNRPA1, MCL1, USP8, MED15, EIF3C, TRIM21, RPL38, PSMA1, IRF4, IKZF1, CDK11, WEE1, RPL27, CDK11B, RAB11A, MAF, NDC80, CCND2, COPB2, CDK11A, IKZF3, EFNA2, KIF11, HIP1, RRM1, CKAP5, PIM2, RSF1, NUF2, CUL9, CARS, TRIM68, USP36, WBSCR22, AGTRAP, TUBGCP6, TPMT]

Closeness Centrality

Top gene: UBC

Number of input genes for closeness centrality: 19

Input genes for closeness centrality: [NFKB1, TUBB, NOS2, NR3C1, PSMB5, TNF, PTGS2, ANXA1, PSMB1, PSMB2, PSMB9, TUBA4A, NR1I2, XPO1, TOP2A, NOLC1, PSMB10, FGFR2, NR0B1]

Number of essential genes for closeness centrality: 46

Essential genes for closeness centrality: [RELA, NFKB1, PLK1, UBB, IKBKB, NFKB2, PSMA4, SNW1, PSMA3, EIF4A3, PSMC5, RELB, PSMC3, PSMC4, PSMA6, PRPF8, AURKB, SF3A1, XPO1, NDC80, CSNK1A1, COPB2, TPMT, TRIM21, IRF4, WEE1, RSF1, NUF2, IKZF1, IK, KIF11, MCL1, MED14, RAB11A, TNK2, SNRPA1, CKAP5, EIF3C, MED15, USP8, RPL38, IKZF3, PIM2, RPL27, RRM1, CDK11B]

Betweenness Centrality

Top gene: UBC

Number of input genes for betweenness centrality: 23

Input genes for betweenness centrality: [NFKB1, NR3C1, XPO1, TUBB, FGFR2, NOS2, TNF, TNFSF11, ANXA1, PSMB1, TOP2A, NR1I2, NOLC1, TUBA4A, CDH5, PTGS2, PSMB5, NR0B1, GSR, HSD11B1, PSMB2, PSMB8, PSMB10]

Number of essential genes for betweenness centrality: 47

Essential genes for betweenness centrality: [RELA, SNW1, IKBKB, EIF4A3, PLK1, NFKB1, AURKB, XPO1, UBB, PSMA3, CSNK1A1, PSMC5, TNK2, PRPF8, SF3A1, USP8, MED14, EIF3C, NFKB2, MCL1, IK, PSMC3, MED15, PSMA6, RPL38, IRF4, RELB, RAB11A, TRIM21, PSMA4, PSMC4, SNRPA1, IKZF1, COPB2, NDC80, EFNA2, WEE1, CDK11, HIP1, CDK11B, MAF, RPL27, CDK11A, IKZF3, CCND2, CKAP5, USP36]

Harmonic Centrality

Top gene: UBC

Number of input genes for harmonic centrality: 18

Input genes for harmonic centrality: [NFKB1, NOS2, TUBB, NR3C1, TNF, PSMB5, PTGS2, PSMB1, XPO1, PSMB2, ANXA1, PSMB9, TUBA4A, NR1I2, TOP2A, NOLC1, PSMB10, FGFR2]

Number of essential genes for harmonic centrality: 36

Essential genes for harmonic centrality: [RELA, NFKB1, UBB, PLK1, IKBKB, EIF4A3, PSMA3, NFKB2, PSMC5, PSMA4, SNW1, PSMC3, PSMC4, PSMA6,

RELB, AURKB, PRPF8, SF3A1, XPO1, CSNK1A1, NDC80, TRIM21, COPB2, IRF4, TPMT, WEE1, IK, IKZF1, MCL1, SNRPA1, MED14, MED15, TNK2, RSF1, NUF2, EIF3C]

Degree Prestige

Top gene: RELA

Number of input genes for degree prestige: 21

Input genes for degree prestige: [NFKB1, NOS2, NR3C1, TUBB, TNF, XPO1, PSMB5, PSMB1, PTGS2, TOP2A, PSMB9, FGFR2, PSMB2, NOLC1, TUBA4A, ANXA1, PSMB8, NR1I2, TNFSF11, PSMB10, CDH5]

Number of essential genes for degree prestige: 46

Essential genes for degree prestige: [RELA, EIF4A3, IKBKB, UBB, NFKB1, PSMA3, PLK1, PSMC5, XPO1, PSMC3, AURKB, PSMC4, SNW1, PSMA6, NFKB2, PRPF8, PSMA4, SF3A1, RELB, MCL1, CSNK1A1, IK, MED15, SNRPA1, MED14, TNK2, IKZF1, USP8, EIF3C, IRF4, TRIM21, RPL27, RPL38, CCND2, RAB11A, WEE1, MAF, NDC80, IKZF3, CDK11B, COPB2, KIF11, PSMA1, CDK11, RRM1, CKAP5]

Eigenvector Prestige

Top gene: PSMC5

Number of input genes for eigenvector prestige: 17

Input genes for eigenvector prestige: [PSMB5, NOS2, PSMB9, PSMB1, PSMB2, NFKB1, PSMB8, PSMB10, NR3C1, TUBB, PTGS2, TNF, XPO1, TUBA4A, ANXA1, NR1I2, TOP2A]

Number of essential genes for eigenvector prestige: 21

Essential genes for eigenvector prestige: [PSMC5, PSMC3, PSMA3, PSMC4, PSMA4, RELA, PSMA6, PLK1, NFKB1, UBB, IKBKB, NFKB2, EIF4A3, AURKB, RELB, SNW1, PSMA1, XPO1, SF3A1, PRPF8, CSNK1A1]

Katz Prestige

Top gene: CUL9

Number of input genes for Katz prestige: 0

Input genes for Katz prestige: []

Number of essential genes for Katz prestige: 16

Essential genes for Katz prestige: [CUL9, SNRPA1, CKAP5, MAF, CDK11A, HIP1, CDK11, PSMC4, UBB, PSMA1, MCL1, CARS, RPL27, RRM1, IKZF1, EFNA2]

A.2 MM-0038 Network

Degree Centrality

Top gene: RELA

Number of input genes for degree centrality: 21

Input genes for degree centrality: [NFKB1, NR3C1, TUBB, XPO1, NOS2, TNF, PSMB1, PSMB5, PSMB2, TOP2A, TUBA4A, FGFR2, PSMB8, PSMB9, ANXA1, PTGS2, NR1I2, PSMB10, NOLC1, TNFSF11, CDH5]

Number of essential genes for degree centrality: 45

Essential genes for degree centrality: [RELA, SNW1, NFKB1, IKBKB, EIF4A3, PLK1, PSMA3, UBB, AURKB, PSMC5, XPO1, PSMA6, PSMC3, PSMA4, PRPF8, PSMC4, NFKB2, CSNK1A1, SF3A1, RELB, TNK2, MED14, IK, SNRPA1, EIF3C, MCL1, USP8, MED15, TRIM21, IRF4, RPL38, PSMA1, IKZF1, CDK11, RPL27, WEE1, RAB11A, CDK11B, MAF, NDC80, CCND2, COPB2, EFNA2, CDK11A, IKZF3]

Closeness Centrality

Top gene: UBC

Number of input genes for closeness centrality: 19

Input genes for closeness centrality: [NFKB1, TUBB, NOS2, NR3C1, PSMB5, ANXA1, TNF, XPO1, PSMB1, TUBA4A, PTGS2, PSMB2, PSMB9, TOP2A, NR1I2, NOLC1, PSMB10, NR0B1, FGFR2]

Number of essential genes for closeness centrality: 45

Essential genes for closeness centrality: [RELA, PLK1, IKBKB, UBB, NFKB1, NFKB2, PSMA3, EIF4A3, PSMA4, PSMC5, SNW1, PSMC3, PRPF8, PSMC4, RELB, AURKB, PSMA6, XPO1, SF3A1, TRIM21, CSNK1A1, COPB2, NDC80, RSF1, IK, WEE1, IRF4, TPMT, IKZF1, KIF11, NUF2, EIF3C, MED14, RPL27, MCL1, MED15, IKZF3, USP8, RPL38, CKAP5, RRM1, SNRPA1, TNK2, CDK11B, CARS]

Betweenness Centrality

Top gene: UBC

Number of input genes for betweenness centrality: 19

Input genes for betweenness centrality: [NFKB1, NR3C1, XPO1, TUBB, NOS2, TNF, FGFR2, TNFSF11, PSMB1, TOP2A, ANXA1, NR1I2, NOLC1, TUBA4A, PTGS2, PSMB5, CDH5, GSR, NR0B1]

Number of essential genes for betweenness centrality: 44

Essential genes for betweenness centrality: [RELA, SNW1, IKBKB, EIF4A3, PLK1, NFKB1, AURKB, XPO1, UBB, PSMA3, CSNK1A1, PSMC5, PRPF8, TNK2, SF3A1, EIF3C, MED14, USP8, NFKB2, MCL1, PSMC3, IK, RAB11A, MED15, PSMA6, IRF4, RPL38, TRIM21, RELB, PSMC4, SNRPA1, PSMA4, IKZF1, COPB2, CDK11, EFNA2, CDK11B, NDC80, WEE1, MAF, RPL27, HIP1, IKZF3, CDK11A]

Harmonic Centrality

Top gene: UBC

Number of input genes for harmonic centrality: 18

Input genes for harmonic centrality: [NFKB1, TUBB, NOS2, NR3C1, TNF, PSMB5, XPO1, PSMB1, PTGS2, ANXA1, TUBA4A, PSMB2, PSMB9, TOP2A, NR1I2, NOLC1, PSMB10, FGFR2]

Number of essential genes for harmonic centrality: 35

Essential genes for harmonic centrality: [RELA, IKBKB, PLK1, UBB, NFKB1, EIF4A3, PSMA3, NFKB2, PSMC5, PSMA4, PSMC3, SNW1, PSMC4, PRPF8, PSMA6, AURKB, RELB, XPO1, SF3A1, TRIM21, CSNK1A1, IK, NDC80, COPB2, IRF4, WEE1, TPMT, RSF1, IKZF1, MCL1, SNRPA1, MED15, MED14, EIF3C, RPL27]

Degree Prestige

Top gene: RELA

Number of input genes for degree prestige: 21

Input genes for degree prestige: [NFKB1, NOS2, NR3C1, TUBB, TNF, XPO1, PSMB5, PSMB1, TOP2A, PTGS2, PSMB9, FGFR2, TUBA4A, PSMB2, NOLC1, ANXA1, PSMB8, NR1I2, TNFSF11, PSMB10, CDH5]

Number of essential genes for degree prestige: 45

Essential genes for degree prestige: [RELA, EIF4A3, IKBKB, UBB, NFKB1, PSMA3, PLK1, PSMC5, XPO1, PSMC3, AURKB, PSMC4, PSMA6, SNW1, PRPF8, NFKB2, PSMA4, SF3A1, RELB, CSNK1A1, MCL1, IK, MED15, SNRPA1, TNK2, EIF3C, MED14, USP8, RPL27, IKZF1, IRF4, TRIM21, RPL38, RAB11A, WEE1, CCND2, MAF, NDC80, COPB2, IKZF3, CDK11B, CDK11, KIF11, PSMA1, CKAP5]

Eigenvector Prestige

Top gene: PSMC5

Number of input genes for eigenvector prestige: 17

Input genes for eigenvector prestige: [PSMB5, NOS2, PSMB9, PSMB1, PSMB2, NFKB1, PSMB8, PSMB10, TUBB, NR3C1, TNF, PTGS2, XPO1, TUBA4A, ANXA1, TOP2A, NR1I2]

Number of essential genes for eigenvector prestige: 20

Essential genes for eigenvector prestige: [PSMC5, PSMC3, PSMA3, PSMC4, PSMA4, RELA, PSMA6, PLK1, NFKB1, IKBKB, UBB, NFKB2, EIF4A3, AURKB, RELB, SNW1, XPO1, SF3A1, PSMA1, PRPF8]

Katz Prestige

Top gene: DACH1

Number of input genes for Katz prestige: 6

Input genes for Katz prestige: [PSMB2, GSR, NOS2, CRBN, PSMB9, NR0B1]

Number of essential genes for Katz prestige: 10

Essential genes for Katz prestige: [PSMA1, COPB2, MAF, IKZF3, CDK11, PSMA3, SNRPA1, WEE1, PIM2, UBB]

A.3 MM-0191 Network

Degree Centrality

Top gene: SNW1

Number of input genes for degree centrality: 20

Input genes for degree centrality: [NFKB1, NR3C1, TUBB, XPO1, NOS2, TNF, PSMB1, PSMB5, TUBA4A, PSMB2, TOP2A, FGFR2, ANXA1, PSMB8, PSMB9, PTGS2, NR1I2, NOLC1, PSMB10, TNFSF11]

Number of essential genes for degree centrality: 42

Essential genes for degree centrality: [SNW1, RELA, IKBKB, EIF4A3, NFKB1, PLK1, UBB, PSMA3, AURKB, PSMC5, XPO1, PSMA6, PSMC3, CSNK1A1, PSMA4, NFKB2, PRPF8, PSMC4, SF3A1, TNK2, RELB, MED14, IK, SNRPA1, MCL1, USP8, EIF3C, RPL38, IRF4, MED15, TRIM21, PSMA1, RPL27, IKZF1, CDK11, RAB11A, WEE1, CDK11B, MAF, NDC80, CCND2, COPB2]

Closeness Centrality

Top gene: UBC

Number of input genes for closeness centrality: 20

Input genes for closeness centrality: [NFKB1, TUBB, NOS2, NR3C1, XPO1, PSMB5, TUBA4A, PSMB1, PTGS2, ANXA1, PSMB9, PSMB2, TNF, TOP2A, NR1I2, NOLC1, PSMB10, FGFR2, CRBN, NR0B1]

Number of essential genes for closeness centrality: 45

Essential genes for closeness centrality: [RELA, PLK1, UBB, NFKB1, IKBKB, PSMA3, NFKB2, EIF4A3, PSMA4, PSMC5, AURKB, XPO1, SNW1, PSMC3, RELB, PSMC4, PSMA6, PRPF8, SF3A1, NDC80, CSNK1A1, IRF4, COPB2, TRIM21, WEE1, TPMT, RRM1, TNK2, KIF11, MED14, RSF1, MED15, RPL38, NUF2, MCL1, IK, USP8, CKAP5, SNRPA1, EIF3C, CDK11B, IKZF3, IKZF1, RAB11A, RPL27]

Betweenness Centrality

Top gene: UBC

Number of input genes for betweenness centrality: 17

Input genes for betweenness centrality: [NFKB1, NR3C1, XPO1, TUBB, FGFR2, TNF, NOS2, ANXA1, PSMB1, NR1I2, TNFSF11, TUBA4A, TOP2A, CDH5, PTGS2, NOLC1, PSMB5]

Number of essential genes for betweenness centrality: 40

Essential genes for betweenness centrality: [RELA, SNW1, IKBKB, EIF4A3, PLK1, NFKB1, AURKB, UBB, XPO1, CSNK1A1, PSMA3, PSMC5, TNK2, PRPF8, USP8, SF3A1, MCL1, NFKB2, MED14, EIF3C, IRF4, IK, PSMC3, PSMA6, RELB, RAB11A, MED15, RPL38, TRIM21, SNRPA1, IKZF1, PSMA4, PSMC4, COPB2, EFNA2, CDK11, WEE1, NDC80, CDK11B, MAF]

Harmonic Centrality

Top gene: UBC

Number of input genes for harmonic centrality: 19

Input genes for harmonic centrality: [NFKB1, TUBB, NOS2, NR3C1, XPO1, PSMB5, TNF, PTGS2, PSMB1, TUBA4A, PSMB2, PSMB9, ANXA1, TOP2A, NR1I2, NOLC1, PSMB10, FGFR2, CRBN]

Number of essential genes for harmonic centrality: 38

Essential genes for harmonic centrality: [RELA, PLK1, UBB, IKBKB, NFKB1, PSMA3, EIF4A3, NFKB2, PSMC5, PSMA4, XPO1, AURKB, PSMC3, SNW1, PSMC4, PSMA6, RELB, PRPF8, SF3A1, CSNK1A1, NDC80, IRF4, TRIM21, COPB2, WEE1, TPMT, MCL1, TNK2, SNRPA1, IK, MED15, MED14, RPL38, RRM1, KIF11, RPL27, EIF3C, USP8]

Degree Prestige

Top gene: RELA

Number of input genes for degree prestige: 20

Input genes for degree prestige: [NOS2, NFKB1, NR3C1, TUBB, XPO1, TNF, PSMB1, PSMB5, FGFR2, TOP2A, PTGS2, TUBA4A, PSMB9, PSMB2, NOLC1, ANXA1, PSMB8, NR1I2, TNFSF11, PSMB10]

Number of essential genes for degree prestige: 43

Essential genes for degree prestige: [RELA, EIF4A3, UBB, IKBKB, NFKB1, PSMA3, PLK1, PSMC5, XPO1, PSMC3, AURKB, PSMC4, PRPF8, SNW1, NFKB2, PSMA6, PSMA4, SF3A1, RELB, MCL1, CSNK1A1, IK, TNK2, RPL27, SNRPA1, EIF3C, MED15, MED14, USP8, IRF4, RPL38, IKZF1, TRIM21, RAB11A, CCND2, WEE1, MAF, NDC80, COPB2, IKZF3, CDK11, KIF11, CDK11B]

Eigenvector Prestige

Top gene: UBC

Number of input genes for eigenvector prestige: 16

Input genes for eigenvector prestige: [PSMB5, NOS2, PSMB9, PSMB1, PSMB2, NFKB1, PSMB8, PSMB10, TUBB, NR3C1, PTGS2, TNF, XPO1, TUBA4A, ANXA1, TOP2A]

Number of essential genes for eigenvector prestige: 20

Essential genes for eigenvector prestige: [PSMC5, PSMA3, PSMC3, PSMC4, PSMA4, PSMA6, RELA, PLK1, NFKB1, UBB, IKBKB, NFKB2, EIF4A3, AURKB, XPO1, RELB, SNW1, SF3A1, PSMA1, PRPF8]

Katz Prestige

Top gene: UBE2N

Number of input genes for Katz prestige: 6

Input genes for Katz prestige: [PTGS2, GSR, CRBN, PSMB2, HSD11B1, NOS2]

Number of essential genes for Katz prestige: 8

Essential genes for Katz prestige: [KIF11, COPB2, IKZF3, NFKB2, RPL27, PSMA1, CDK11, CCND2]